



Visualisation stéréoscopique et interactive de structures en communautés dans des graphes

Nicolas Greffard

► To cite this version:

Nicolas Greffard. Visualisation stéréoscopique et interactive de structures en communautés dans des graphes. Interface homme-machine [cs.HC]. Université de Nantes, 2013. Français. NNT: . tel-01072368

HAL Id: tel-01072368

<https://theses.hal.science/tel-01072368>

Submitted on 8 Oct 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse de Doctorat

Nicolas GREFFARD

*Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Université de Nantes
sous le label de l'Université de Nantes Angers Le Mans*

Discipline : Informatique et applications
Laboratoire : Laboratoire d'informatique de Nantes-Atlantique (LINA)

Soutenue le 05 Novembre 2013

École doctorale : ED 503 (STIM)

Visualisation Stéréoscopique et Interactive de Structures en Communautés dans des Graphes

JURY

Rapporteurs : **M. Guy MELANCON**, Professeur des Universités, Université Bordeaux 1
M. Gilles VENTURINI, Professeur des Universités, Université Polytechnique de Tours

Examineurs : **M. Benoît OTJACQUES**, Directeur scientifique adjoint, Centre G. Lippmann
M. Fabien PICAROUGNE, Maître de conférences (Co-encadrant), Université de Nantes

Directrice de thèse : **M^{me} Pascale KUNTZ**, Professeur des Universités, Université de Nantes

Remerciements

Bien que cette thèse soit à l'origine un défi personnel, je n'aurais jamais pu la mener à bien sans un certain nombre de personnes que je tiens à remercier.

Avant tout, je tiens à exprimer ma plus profonde gratitude à Mme Pascale Kuntz, ma directrice de thèse et M. Fabien Picarougne, mon encadrant, qui ont su m'accompagner et me guider tout au long de cette thèse. Sans leur expertise théorique, technique et pratique, cette thèse n'aurait certainement jamais abouti.

Je tiens également à remercier M. Gilles Venturini et M. Guy Melancon qui m'ont fait l'honneur d'accepter d'être rapporteurs de cette thèse. Ainsi que M. Benoît Otjacques qui a accepté d'être examinateur lors de ma soutenance.

J'ai eu la chance de travailler au sein de l'équipe CONnaissances et Décisions à l'école polytechnique de l'université de Nantes dont je tiens à remercier tous les membres et personnels qui ont facilité le déroulement de cette thèse. Plus particulièrement je remercie mes collègues qui au détour d'un couloir ou autour d'un café ont grandement facilité le quotidien.

Un remerciement particulier à M. Abdelghani Hadj-Rabia sans qui je ne serais jamais arrivé jusqu'ici.

Fussent-ils proches ou plus éloignés, mes amis ont toujours été là. Sans aucun ordre particulier, merci à Maxence, Matthieu, Charles, Yvonne, Mathilde, Romain, François, Antoine, Vincent, Stéphane, Antoine, Charlotte, David, Marie, Clémentine, Maël et Philippe. Sans vous, je n'y serais jamais arrivé.

Enfin, je remercie ma famille qui m'a toujours soutenu et surtout Guillemine.

Table des matières

1	Introduction	13
1.1	Contexte scientifique	14
1.2	Contributions	17
1.3	Organisation du manuscrit	18
2	Visualisation 3D et Perception	21
	Introduction	22
2.1	Perception de la profondeur	23
2.1.1	Mécanismes à l'origine de la perception de la profondeur	23
2.1.2	Importance relative des mécanismes de perception de la pro- fondeur	30
2.1.3	Discussion	31
2.2	Apports et limites de la stéréoscopie	32
2.2.1	Etudes favorables à la stéréoscopie	33
2.2.2	Etudes défavorables à la stéréoscopie	36
2.2.3	Discussion	37
2.3	La visualisation immersive de graphes	37
2.3.1	La visualisation 3D stéréoscopique de graphes	38
2.3.2	Visualisation hautement immersive de graphes	39
2.3.3	Discussion	40
3	Fouille visuelle de graphes	41
	Introduction	42
3.1	Tâches en fouille de graphes	43
3.1.1	Typologie générale	44
3.1.2	Détection de communautés	46
3.2	Tracés de graphes	47
3.2.1	Modèles de représentation et algorithmes de tracé	47
3.2.2	Tracés de type force et ressort	49
3.2.3	Tracés 3D monoscopique	51
3.3	Interactions en fouille de graphes	52

4	TempoSpring	55
	Introduction	56
4.1	Architecture générale	56
4.1.1	Algorithme de tracé	58
4.1.2	Optimisation des calculs	61
4.1.3	Implémentation de la stéréoscopie	63
4.2	Mécanismes d'interaction	63
4.3	Graphes Dynamiques	67
4.3.1	Présentation des principales méthodes d'intégration de la dynamique	68
4.3.2	La dynamique dans <i>TempoSpring</i>	72
5	Détection de communautés en 3D	75
	Introduction	76
5.1	Hypothèses	76
5.2	Protocole Expérimental	77
5.2.1	Matériel	78
5.2.2	Base de Données de Graphes	79
5.2.3	Participants	79
5.2.4	Procédure Expérimentale	81
5.3	Résultats	82
5.3.1	Qualité de la détection de communautés	82
5.3.2	Temps de Réponse	83
5.3.3	Perception des Participants	83
5.3.4	Discussion	85
5.4	Deuxième expérience	86
5.4.1	Améliorations du protocole expérimental	86
5.4.2	Résultats	87
	Discussion	88
6	Etude des interactions	89
	Introduction	90
6.1	Observations et hypothèses	90
6.2	Mesures classiques d'interactions	91
6.3	Indicateurs de mouvements	94
6.4	Analyse des Mouvements	97
6.4.1	Approche exploratoire : classification non supervisée	97
6.4.2	Analyse statistique des indicateurs	98
6.4.3	Discussion	100
6.5	Analyse des points de vue	101

6.5.1	Carte de chaleur 3D	101
6.5.2	Résultats	102
	Discussion	103
7	Vers de nouveaux modes interactifs	107
	Introduction	108
7.1	Interactions dans les logiciels de fouille	109
7.2	Déclenchement des interactions dans <i>TempoSpring</i>	110
7.3	Interface mains libres dans <i>TempoSpring</i>	112
7.3.1	Principes généraux	112
7.3.2	Automate	113
7.3.3	Interactions multimodales	116
	Conclusion	117
8	Conclusion et perspectives	119
8.1	Conclusion	120
8.2	Perspectives	122

Liste des tableaux

3.1	Typologie de tâches proposée par [4]	44
5.1	Implémentation des mécanismes de perception	78
5.2	Erreur dans la détection en fonction de la complexité	83
5.3	Erreur dans la détection en fonction du nombre de communautés	84
5.4	Ecart type de l'erreur dans la détection de communautés	84
5.5	Temps de réponse moyen	85
5.6	Résultats subjectifs	85
6.1	Statistiques pour l'indicateur D_{tot}	98
6.2	Statistiques pour l'indicateur RA	98
6.3	Statistiques pour l'indicateur $\overline{C_{ourb}}$	99
6.4	Statistiques pour l'indicateur P_{tot}	99
6.5	Statistiques pour l'indicateur RP	99
6.6	Statistiques pour l'indicateur \overline{P}	100
6.7	Statistiques pour l'indicateur $\overline{\Delta P}$	100

Table des figures

2.1	Illustration de l’accommodation	24
2.2	Illustration de la convergence	24
2.3	Illustration de la disparité binoculaire	25
2.4	Illustration du parallax	26
2.5	Illustration de l’occlusion	27
2.6	Illustration de la hauteur dans le champ de vision	27
2.7	Illustration des tailles relatives	28
2.8	Illustration des densités relatives	29
2.9	Illustration de la perspective atmosphérique	29
2.10	Importance relative des mécanismes de perception	31
2.11	Rendu géométrique	34
2.12	Rendu volumique	34
3.1	Illustration du pipeline de fouille visuelle	42
3.2	Illustration de deux tracés	48
3.3	Illustration d’un tracé de K7	52
4.1	Diagramme UML TempoSpring	57
4.2	Illustration du format de graphes	59
4.3	Illustration de l’octree	62
4.4	Illustration de la projection mono	64
4.5	Illustration de la projection stéréo	65
4.6	Illustration du surlignage	66
4.7	Illustration du filtrage sur le degré des nœuds	67
4.8	Illustration de l’agrégation manuelle	67
4.9	Illustration des résumés statistiques.	70
4.10	Illustration des diagrammes alluviaux.	70
4.11	Illustration dde la dynamique	74
5.1	Photo de l’installation expérimentale	79
5.2	Capture d’exemples de tracés	80
6.1	Capture d’écran d’une vidéo de la deuxième expérimentation	92

6.2	Illustration des données souris	94
6.3	Illustration d'un point de vue	101
6.4	Illustration d'une géode	102
6.5	Exemple de cartes de chaleur 3D	104
7.1	Photo d'un utilisateur de <i>TempoSpring</i>	111
7.2	Illustration du <i>Kinect</i>	112
7.3	Illustration d'un cas d'utilisation	113
7.4	Illustration d'un cas d'utilisation	114
7.5	Illustration d'un cas d'utilisation	114
7.6	Illustration d'un cas d'utilisation	115
7.7	Illustration d'un cas d'utilisation	115
7.8	Illustration d'un cas d'utilisation	116
7.9	Diagramme de l'automate du moteur d'interactions	117
8.1	Illustration d'un graphe du projet GRAPHCOMP	121
8.2	Illustration du plongement spatial	125

Introduction

A picture is worth a thousand words.

—Arthur Brisbane

Sommaire

1.1	Contexte scientifique	14
1.2	Contributions	17
1.3	Organisation du manuscrit	18

1.1 Contexte scientifique

*Il faut imaginer dans sa tête des trucs qu'on appelle sommets, et pour toute paire de sommets soit une arête qui les joint, soit une non-arête qui les laisse sans joint : ceci est un graphe selon Berge [169], un des pionniers de la théorie des graphes. Contrairement à d'autres objets mathématiques, les graphes ont été étroitement liés dès leur origine à des applications. On rapporte que le terme "graphe" a été introduit en 1878 par J.J. Sylvester pour représenter des relations entre des substances chimiques et des composants organiques [187]. Et, les prémisses de l'analyse des réseaux pratiquée aujourd'hui se trouvent vers la même époque dans des travaux sur les réseaux de communications : "Les voies de communication qui sillonnent les pays civilisés, s'infléchissent dans leurs directions et s'entrecoupent mutuellement de manière à recouvrir le territoire d'une sorte de tissu, de filet à mailles inégales, à contours irréguliers. L'expression de **réseau** caractérise donc d'une manière heureuse l'ensemble des voies navigables dont le développement est naturellement très limité, soit aux routes et chemins ordinaires, soit aux chemins de fer."* [123]. Dans le siècle suivant, les sociologues spécialistes des "réseaux sociaux" [127] et les géographes spécialistes des réseaux de transport [112] ont contribué à dresser les fondations de ce que certains nomment une "économie relationnelle" [31].

Les graphes sont donc devenus des modèles privilégiés pour modéliser et étudier des systèmes de relations entre des entités dans des domaines aussi variés que la lutte contre le terrorisme [165], les applications marketing [89], ou l'analyse des réseaux corticaux [34]. L'accessibilité croissante de ces systèmes à des échelles qui défient certainement ce que les pionniers avaient pu imaginer a conduit au développement d'indicateurs permettant de quantifier leurs propriétés structurelles et d'algorithmes de calcul et d'extraction de motifs caractéristiques pour des complexités adaptées à ces changements d'échelles.

A cette analyse métrologique et combinatoire, s'est associé un besoin croissant de visualisation. Claude Berge annonçait déjà que "*Quand on étudie la théorie des graphes, on pourrait très bien parler de graphe en terme de fonction en 0 et 1 (...), mais non, on les traite en forme de figure parce qu'on veut visualiser l'objet, mettre des points pour représenter des sommets ; les arêtes ce sont des lignes continues qu'on dessine sur le plan et ce sont des propriétés d'un type graphique et visuel qu'on étudie (...)*". Les sociologues illustraient leurs premières analyses par des tracés à main levée de graphes afin de mettre en valeur certaines propriétés caractéristiques [140]. Dans cette célèbre étude de Moreno, le tracé du graphe comprenait moins d'une quarantaine de nœuds. Tracer de la sorte le graphe représentant le réseau social *Facebook* comportant plus de 700 millions de nœuds s'avèrerait

impensable. Depuis les travaux précurseurs de Tutte [193] dans les années 60, de nombreuses approches de tracés de graphes ont été proposées (e.g. [52, 143]) et implémentées dans des outils logiciels (e.g. [96, 109]).

Si une partie importante des travaux s’est concentrée -et continue de le faire- sur les problèmes combinatoires et algorithmiques posés par les dessins de graphes, les besoins des utilisateurs sont devenus une préoccupation croissante. On parle désormais de “fouille visuelle” de réseaux et plus généralement de “visual analytics”. L’utilisateur est placé au cœur de l’analyse via des supports visuels interactifs qui lui permettent de piloter le processus de fouille. Comme l’a dit Keim qui a fortement contribué au développement de cette approche dans la communauté de Visualisation de l’Information : “*visual analytics is more than visualization and can rather be seen as an integrated approach combining visualization, human factors and data analysis*” [113]. Le but de la fouille visuelle est donc de faciliter la recherche, la déduction et l’identification de caractéristiques intéressantes via des systèmes interactifs en exploitant les capacités humaines de perception, d’abstraction et de compréhension des données complexes visualisées.

De nombreuses questions sont soulevées à la fois par la prise en compte des capacités cognitives et perceptives des utilisateurs humains et par les changements d’échelle. Un certain nombre se regroupent sous le terme de “complexité visuelle” [96]. Pour faire face aux challenges d’aujourd’hui trois approches principales, rappelées dans la conférence invitée de P. Eades lors de la 18^e conférence Graph Drawing [61], sont proposées :

- L’agrégation hiérarchique : regrouper les nœuds [5] ou les liens [190] dans des *meta-nœuds* (respectivement *meta-liens*) de manière hiérarchique pour que seuls les plus hauts dans la hiérarchie soient affichés simultanément. Les utilisateurs ont généralement la possibilité de développer ces meta éléments afin de parcourir la hiérarchie dans son ensemble.
- Les interactions : les détails du graphe ne sont révélés que via des interactions spécifiques en utilisant par exemple les métaphores du *Fisheye* [174] ou de la loupe grossissante [151].
- Visualisation 3D : le graphe est affiché en 3 dimensions de façon à pouvoir afficher, de manière interprétable, plus d’éléments tout en occupant le même espace d’affichage qu’une restitution 2D.

On peut supposer que la complexité actuelle des problèmes nécessite certainement une combinaison efficace de ces différentes approches. Cependant, on se heurte alors à une autre question : celle de l’évaluation. Et les études publiées se limitent généralement à l’évaluation d’apports de fonctionnalités assez restreintes et ne couvrant pas l’ensemble des tâches -dont l’identification reste d’ailleurs un

sujet de débat- d'un processus complet d'analyse. Nous ne dérogeons pas à ce positionnement dans cette thèse et nous focalisons sur l'analyse de l'apport de la 3D stéréoscopique dans l'analyse des graphes.

Certains attribuent la découverte de la vision stéréoscopique à Euclide, il y a plus de 2300 ans dans son ouvrage "Optique" [35] dans lequel il établit les préceptes de la perspective. Bien qu'une certaine intuition de la vision stéréoscopique puisse être imputable à ses propos : *"When a sphere is seen by both eyes, if the diameter of the sphere is equal to the straight line marking the distance of the eyes from each other, the whole hemisphere will be seen"* ; il faut en réalité attendre le 19^e siècle et l'invention du stéréoscope par Wheatstone [196] pour que les mécanismes à l'origine de la vision stéréoscopique tels que nous les connaissons aujourd'hui soient découverts. Cet astucieux arrangement de miroirs permet de voir simultanément deux images différentes : une pour chaque œil, et ainsi de percevoir une image en relief.

Depuis, les technologies ont considérablement évolué et il est aujourd'hui possible de visualiser des scènes extrêmement complexes d'images virtuelles restituées en temps réel. Ce développement s'est accompagné d'une baisse du coût de ces technologies, entraînant leur démocratisation et relançant l'intérêt de la communauté scientifique pour les restitutions 3D qui avaient été souvent critiquées dans leurs variantes monoscopiques [61, 41].

L'objectif de cette thèse est d'analyser l'apport de la perception de la profondeur introduite dans un dispositif de restitution stéréoscopique pour la détection de propriétés structurelles dans des graphes. Ici, nous nous sommes focalisés sur une tâche de "haut niveau" eu égard à d'autres tâches précédemment étudiées (e.g. identification de nœuds, de chaînes de longueur restreinte) : la détection de structures en communautés. Cette problématique est majeure en analyse de réseaux sociaux et une large littérature lui a été consacrée (voir [74] pour un récent état de l'art). Il ne s'agit pas dans cette thèse de présenter un antépénultième algorithme permettant d'identifier automatiquement des communautés mais d'analyser l'impact de la 3D dans un processus de détection interactif. Nous nous appuyons donc sur une des approches de tracé les plus utilisées : un algorithme de type force et ressort. D'autres algorithmes plus performants ont été développés (e.g. l'algorithme Lin-Log de [147]) mais celui-là, outre sa popularité, présente l'avantage dans notre cas de permettre des restitutions visuelles en 2D, 3D monoscopique et 3D stéréoscopique sans transformations profondes de l'algorithme qui pourraient entraîner des biais importants dans les comparaisons. Dans ce cadre, deux questions majeures ont retenu notre attention : l'impact de la 3D stéréoscopique dans les performances de détection, et aussi dans les interactions de l'utilisateur sur les graphes. D'une façon

générale, l'analyse des interactions reste un champ largement à explorer en fouille visuelle de données ; et dans le cas particulier des environnements 3D stéréo nous n'avions pas connaissance d'études préalables.

1.2 Contributions

Les expérimentations sur lesquelles nous nous sommes appuyées pour analyser l'impact de la stéréoscopie dans une tâche de détection de structures en communautés dans des graphes ont nécessité le développement préalable d'un environnement logiciel adapté. Ainsi, nous pouvons résumer les contributions principales de notre travail selon deux axes : (i) les contributions expérimentales, et (ii) les contributions logicielles.

(i) Contributions expérimentales Notre principale contribution expérimentale est la comparaison des restitutions 3D monoscopique, 3D stéréoscopique et 2D pour une tâche de détection de communautés dans des graphes de complexité variable. Nos résultats montrent que pour les graphes les plus simples, la 2D a de meilleures performances et des temps de réponses plus courts. *A contrario*, la 3D stéréoscopique est plus performante pour les graphes complexes, mais ses temps de réponses restent plus importants qu'en 2D. La 3D monoscopique est toujours en deçà des autres approches.

En complément des analyses de performances pour la résolution de tâche, nous avons comparé les comportements d'interaction pour les conditions 3D monoscopique et 3D stéréoscopique. Dans les expérimentations les interactions ont été restreintes à des rotations du tracé du graphe autour de son centre de gravité. Nos résultats mettent en évidence des différences significatives selon les deux conditions : en particulier, sous la condition 3D monoscopique, les participants ont tendance à explorer l'espace de visualisation de manière homogène tandis que sous la condition 3D stéréoscopique, ils ont tendance à se focaliser sur certaines zones spécifiques. La quantification des résultats a été restituée par une carte de chaleur retraçant les temps passés par les participants sur les différents points de vue adoptés.

Ces différentes expérimentations ont nécessité la construction d'un protocole expérimental spécifiquement adapté à la détection de tâches de détection de communautés pour des traces de graphes de complexité variable (c'est-à-dire ici avec des probabilités de chevauchement entre classes plus ou moins importantes).

(ii) Contributions logicielles Les contributions expérimentales que nous présentons dans cette thèse reposent sur un environnement logiciel expérimental que nous avons développé : *TempoSpring*. Cet outil permet la visualisation immersive et interactive de réseaux sociaux sous forme de graphes. Développé en utilisant des technologies libres, il permet de visualiser des tracés obtenus avec un algorithme de force et ressort itératif proposant plusieurs mécanismes d’aide à la visualisation. Pour nos expérimentations nous nous sommes placés sur les mêmes échelles que les travaux précédemment publiés pour des tâches élémentaires (quelques centaines de nœuds), mais une optimisation des calculs (via programmation parallèle sur la carte graphique) permet de visualiser des graphes de quelques milliers de nœuds en 3D stéréoscopique en temps réel.

En sus, nous proposons l’implémentation d’une bibliothèque d’interaction mains libres permettant d’utiliser le capteur *Kinect* de *Microsoft* au sein d’un logiciel. Cette bibliothèque est notamment utilisée par *TempoSpring* pour offrir des interactions plus adaptées à une visualisation en station debout devant un grand écran.

1.3 Organisation du manuscrit

Ce manuscrit est organisé comme suit :

Le chapitre 2 présente la perception de la profondeur, qui est la composante majeure de la différence entre visualisation 2D et 3D. Nous commençons par rappeler les principaux mécanismes (oculaires, stéréoscopiques, dynamiques et picturaux) qui interviennent dans ce phénomène. Puis, nous nous focalisons sur la perception stéréoscopique qui est au coeur de nos recherches, et à travers un état de l’art des travaux de la littérature nous tentons d’examiner les apports et les limites de la visualisation stéréoscopique eu égard à la visualisation monoscopique. Cette analyse balaye tout d’abord des problématiques applicatives variées dont beaucoup proviennent du domaine médical, et se concentre en dernière partie sur les travaux encore peu nombreux consacrés à l’apport de la stéréoscopie pour la visualisation de graphes.

Le chapitre 3 introduit les approches les plus répandues en fouille visuelle de graphes et indique celles que nous avons retenues dans notre cadre expérimental. Nous nous appuyons dans un premier temps sur quelques tentatives de typologies de tâches pour identifier les tâches majeures de l’analyse des réseaux sociaux, et nous nous focalisons sur la tâche de détection de communautés qui est celle sur laquelle nous avons centré nos expérimentations. Dans un second temps, nous rappelons les

principaux modèles de restitution visuelle de graphes ainsi que les techniques de tracé les plus populaires avant de nous concentrer sur les tracés de type force et ressort et sur les tracés 3D monoscopiques. Puis nous présentons brièvement quelques interactions parmi les plus répandues dans les outils de visualisation de graphes.

Le chapitre 4 décrit *TempoSpring*, l’environnement expérimental pour la fouille de graphes en 3D que nous avons développé lors de cette thèse. Dans un premier temps nous décrivons l’architecture générale de *TempoSpring* comme l’algorithme de tracé choisi ainsi qu’une description succincte de la façon dont la stéréoscopie est implémentée et des mécanismes d’optimisation des calculs. Dans un second temps, nous présentons les interactions proposées dans *TempoSpring* pour faciliter le processus de fouille visuelle. Nous terminons en introduisant un mécanisme de visualisation dynamique de graphes qui, sans avoir été validée expérimentalement, est intégré à notre outil.

Le chapitre 5 est dédié à deux expériences utilisateur que nous avons conduites pour évaluer l’impact de la 3D stéréoscopique pour une tâche de détection de communautés par rapport à une restitution 3D monoscopique et à une restitution 2D. Ces expériences portent sur des graphes de complexité croissante. Nous énumérons dans un premier temps les hypothèses que nous avons formulées. Ensuite, nous décrivons le protocole expérimental retenu pour la première expérience, ainsi que les résultats obtenus. Puis nous examinons les compléments apportés par la seconde expérience qui semble confirmer, pour d’autres participants et d’autres graphes avec des topologies suffisamment difficiles, le caractère prometteur de la stéréoscopie.

Le chapitre 6 aborde une deuxième expérimentation portant sur l’étude des interactions enregistrées lors de la deuxième expérience utilisateur présentée dans le chapitre précédent. Nous énumérons dans un premier temps, les observations qui nous ont conduits à enregistrer ces interactions et les hypothèses qui en ont été déduites. Ensuite, après un rappel des travaux antérieurs sur l’évaluation des interactions avec un périphérique d’interaction classique, nous abordons le premier axe d’étude : l’analyse des mouvements de la souris. Puis, nous présentons le deuxième axe d’étude : une analyse de “plus haut niveau” décrivant les points de vue adoptés par les utilisateurs.

Le chapitre 7 introduit de nouveaux modes interactifs proposés par notre outil *TempoSpring*. Plus précisément, nous utilisons les capacités du capteur *Kinect* de Microsoft pour identifier certains gestes effectués par les utilisateurs pour déclencher les interactions de *TempoSpring*. Cette fonctionnalité, développée sous forme de bibliothèque indépendante, permet de procéder à la fouille visuelle d’un graphe

sans s'encombrer de périphériques exogènes.

Le chapitre 8 qui clôt ce manuscrit rappelle les principales conclusions de nos travaux et en dresse les perspectives. Celles-ci s'orientent dans trois directions : (i) l'évaluation des nouveaux modes d'interaction mains libres, (ii) l'analyse de l'impact de la 3D stéréoscopique dans une visualisation de graphes dynamiques et (iii) l'intégration dans la visualisation de l'information spatiale associée aux nœuds d'un graphe.

Visualisation 3D et Perception

Don't think of that thing as a screen, think of it as a window, a window through which one looks into a virtual world. The challenge to computer graphics is to make that virtual world look real, sound real, move and respond to interaction in real time, and even feel real.

—Ivan E. Sutherland

Sommaire

Introduction	22
2.1 Perception de la profondeur	23
2.1.1 Mécanismes à l'origine de la perception de la profondeur	23
2.1.2 Importance relative des mécanismes de perception de la profondeur	30
2.1.3 Discussion	31
2.2 Apports et limites de la stéréoscopie	32
2.2.1 Etudes favorables à la stéréoscopie	33
2.2.2 Etudes défavorables à la stéréoscopie	36
2.2.3 Discussion	37
2.3 La visualisation immersive de graphes	37
2.3.1 La visualisation 3D stéréoscopique de graphes	38
2.3.2 Visualisation hautement immersive de graphes	39
2.3.3 Discussion	40

Introduction

La visualisation 3D peut s'intégrer dans le cadre plus large des systèmes immersifs. Depuis les années 90, ces derniers ont connu un regain d'intérêt considérable dans le monde de la visualisation d'information, porté par la curiosité des médias, la culture populaire (films hollywoodiens, romans de science fiction) et surtout la demande industrielle (conception assistée par ordinateur, simulation de prototypes). Cette évolution s'est accompagnée d'un essor technologique et d'une diffusion croissante des environnements de développement en réalité virtuelle. En reprenant les mots de Sutherland [186], Brookes [32] définit la réalité virtuelle comme un *écran ultime* : “ *Don't think of that thing as a screen, think of it as a window, a window through which one looks into a virtual world. The challenge to computer graphics is to make that virtual world look real, sound real, move and respond to interaction in real time, and even feel real*”.

La qualité d'un tel système se mesure le plus souvent en fonction du “degré de présence ressenti” par ses utilisateurs [206]. Ce sentiment de présence est associé à un état psychologique représentant la sensation d'être “physiquement présent” dans la scène. Dans le cadre de la réalité virtuelle, cette scène est une simulation informatique, mais cette sensation peut être ressentie lors de la lecture d'un livre par exemple. Elle se caractérise par une forte attention de l'utilisateur, des réactions à des événements ayant lieu dans cet environnement ainsi qu'une réminiscence d'événements passés [106].

Ce phénomène peut être favorisé via l'immersion des utilisateurs dans un environnement virtuel. L'immersion est souvent définie comme un état physique permettant d'isoler l'utilisateur du monde extérieur pour rendre les sensations associées à l'expérience vécue proche de celles qui apparaîtraient en situation réelle [27]. Dans le cadre de la visualisation d'informations, des études suggèrent que les systèmes immersifs peuvent améliorer les processus de perception et d'interprétation des données présentées [125, 157].

Dans ce chapitre, nous nous restreignons à une seule des dimensions mises en jeu dans un système immersif : la perception de la profondeur, composante principale de la différence entre visualisation 2D et 3D. Tout d'abord, nous rappelons les principes majeurs qui interviennent dans la perception de la profondeur : mécanismes oculaires, la disparité binoculaire, le parallaxe et les mécanismes picturaux, et nous tentons de cerner l'importance relative de chacun de ces mécanismes. Ensuite, nous discutons à travers des études expérimentales de la littérature des apports spécifiques mais aussi des limites de la stéréoscopie par rapport à la monoscopie. Nous finissons ce chapitre en nous focalisant sur la fouille visuelle de graphes dont

les rares études portant sur la stéréoscopie semblent encourageantes.

2.1 Perception de la profondeur

Nous vivons dans un espace tridimensionnel et une longue période d'évolution nous a doté d'un système visuel complexe, nous permettant de percevoir cet espace grâce, en outre, aux *stimuli* visuels que nous recevons. De nombreuses études en physiologie et en sciences cognitives ont été dédiées à l'exploration des mécanismes responsables de la perception de la profondeur, composante essentielle de notre perception tridimensionnelle.

Dans ce chapitre, nous tentons d'établir, à partir des travaux de la littérature, une liste des mécanismes mis en jeu dans notre vision tridimensionnelle ; nous nous focalisons ici sur la perception de la profondeur, et nous n'abordons pas explicitement les autres aspects de la perception spatiale qui ont été largement traités [162, 114]. Nous rappelons ensuite que ces mécanismes sont d'une part d'une précision variable en fonction de la distance entre l'observateur et l'objet observé et d'autre part qu'ils sont complémentaires.

2.1.1 Mécanismes à l'origine de la perception de la profondeur

La perception de la profondeur est le résultat de la combinaison de différents mécanismes qui agissent de concert pour permettre à notre cerveau d'inférer des informations quant à la distance d'objets le long de l'axe optique, c'est à dire depuis l'œil jusqu'à l'horizon [49]. L'importance relative de ces mécanismes dépend principalement de la distance entre l'observateur et la scène observée. En effet, certains mécanismes jouent un rôle primordial pour de courtes distances, mais un rôle secondaire, voire très limité selon les connaissances actuelles, pour de longues distances.

Cette section liste les 9 mécanismes principaux qui interviennent dans notre perception de la profondeur appelés parfois "indices de profondeur" (de l'anglais *depth cues*) qui peuvent être d'origine oculaire, stéréoscopique, dynamique ou picturale.

2.1.1.1 Mécanismes oculaires

Les mécanismes oculaires sont directement liés aux changements d'état de l'œil et de ses composants.

Accommodation L'accommodation correspond aux changements de formes du cristallin dans l'œil qui permettent de garder une image rétinienne nette lorsque l'on regarde un objet. Duale, d'autres objets plus proches ou plus éloignés



FIGURE 2.1 – Illustration de l’accommodation. En optique instrumentale, on parle plutôt de mise au point mais le principe sous-jacent est identique. Sur cette photo, la mise au point est réglée sur l’oiseau au premier plan, le deuxième plan apparaît donc comme flou. *Image provenant de flickr sous licence Creative Commons de l’utilisateur pumcus.*

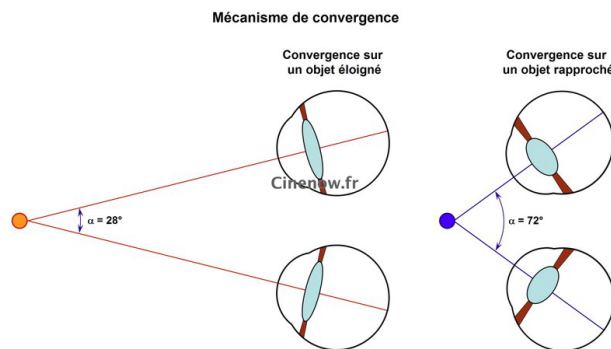


FIGURE 2.2 – Illustration de la convergence. Plus un objet est éloigné de l’observateur, plus l’angle entre les deux axes optiques tend vers 0. *Image provenant de <http://www.cinenow.fr/>.*

sont perçus comme étant flous. Ce mécanisme permet d’inférer une information de nature ordinaire quant à la distance qui sépare les objets. Il est illustré sur la figure 2.1.

Convergence La convergence est mesurée par l’angle entre les axes optiques des deux yeux. Chacun de ses axes a pour origine la fovéa de l’œil et est dirigé vers le point de fixation. Plus un objet se trouve proche de l’observateur, plus l’angle est important. A l’inverse, lorsqu’un observateur fixe un objet très éloigné, cet angle tend vers 0. Ce mécanisme est illustré sur la figure 2.2.

2.1.1.2 Mécanisme stéréoscopique

La distance entre les deux yeux humains nous permet de percevoir le monde depuis deux points de vue légèrement différents. Ceux-ci entraînent une disparité binoculaire qui comporte de l’information sur la distance séparant des objets sur l’axe optique.

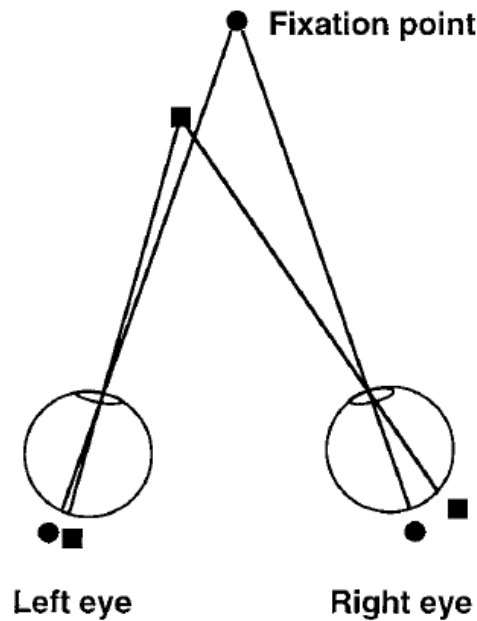


FIGURE 2.3 – Illustration de la disparité binoculaire. En considérant un point de fixation et un autre point distinct, nous observons que les distances angulaires entre leurs projections rétiniennes sont différentes.

Disparité binoculaire La disparité binoculaire provient de la différence relative des projections rétiniennes d'un même objet [8]. Comme on peut le voir sur la figure 2.3, en considérant un point de fixation, les projections rétiniennes d'un deuxième point sont à des distances angulaires différentes de la projection du point de fixation. En connaissant l'angle de convergence et la distance au point de fixation, cette différence permet d'inférer une information absolue [124] quant à la distance qui sépare réellement les deux points. Si la distance au point de fixation est inconnue, l'information est ordinale.

2.1.1.3 Mécanisme dynamique : parallaxe

Le système visuel humain contribue à la perception et l'interprétation des mouvements et de la dynamique du monde qui nous entoure. Contrairement aux autres mécanismes présentés dans ce chapitre, le parallaxe n'est pas mis en œuvre sur une image statique mais dans un contexte dynamique.

Le parallaxe, communément qualifié “de mouvement” s'applique dans deux situations distinctes : un observateur “fixe” observe des objets en mouvements, ou un observateur en mouvement observe des objets fixes. Dans ces deux cas, si les objets sont à des distances différentes, leurs vitesses de déplacement sont perçues comme étant différentes. L'analogie d'un train en marche (voir figure 2.4) illustre ce ressenti. Ce mécanisme permet de déduire une information absolue de la distance



FIGURE 2.4 – Illustration du parallaxe. En photographie, un mouvement rapide d’un objet photographié entraîne un flou cinétique. Sur cette photo prise depuis un train en marche, les arbustes du premier plan sont extrêmement flous tandis que les arbres plus éloignés apparaissent très nettement. *Image provenant de flickr sous licence Creative Commons de l’utilisateur Michael of Scott.*

entre objets à moins de 5m, et une information ordinale au-delà [108].

2.1.1.4 Mécanismes picturaux

Les autres mécanismes sont associés à l’interprétation des images perçues via le système visuel. Ils dépendent des relations entre les tailles, les densités, et les positions des objets reconnus via le système visuel.

Occlusion L’occlusion survient lorsqu’un objet cache intégralement ou partiellement un autre objet. La reconnaissance des objets et de leurs contours permet alors d’extrapoler l’ordre dans lequel ils sont positionnés sur l’axe optique. Par nature, cet indice offre donc une information ordinale. C’est le mécanisme principal de perception de la profondeur, quelle que soit la distance [50]. Il est illustré par la figure 2.5.

Hauteur dans le champ de vision La hauteur dans le champ de vision mesure les relations entre les hauteurs des projections rétinienne des bases d’objets, c’est à dire le point d’intersection entre l’objet et le sol sur lequel il repose. Ce mécanisme requiert la présence d’un sol (les objets reposent dessus), de la gravité (les objets ne flottent pas) et de l’absence d’un plafond. La base d’un objet éloigné se trouve plus haut dans le champ de vision de l’observateur que la base d’un objet se trouvant à sa proximité. La hauteur dans le champ de vision fournit une information de nature ordinale ; son principe est illustré sur la figure 2.6.

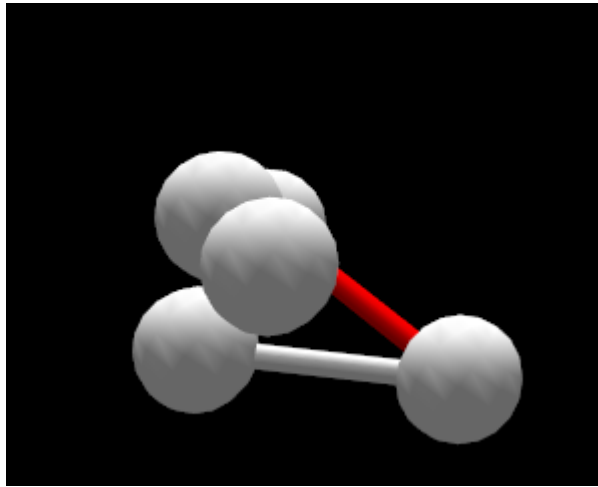


FIGURE 2.5 – Illustration de l’occlusion. Ce graphe très simple illustre l’importance de l’occlusion dans la perception de la profondeur. L’ordonnancement des 3 nœuds situés en haut à gauche apparaît instantanément. Mais sans autre indice de profondeur, il est difficile de déterminer si le nœud de droite est plus proche ou au contraire plus éloigné de l’observateur que les autres nœuds.



FIGURE 2.6 – Illustration de la hauteur de le champ de vision. Cette photo d’un champ de maïs illustre ce phénomène. La base d’un plan est le point d’intersection entre le sol et la tige (par exemple celles marquées d’un trait noir). Nous pouvons directement constater que la hauteur de la base dans l’image augmente avec la distance par rapport à l’observateur. *Image provenant de flickr sous licence Creative Commons de l’utilisateur lovestruck.*



FIGURE 2.7 – Illustration des tailles relatives. Les voitures éloignées apparaissent beaucoup plus petites que les voitures proches alors qu’elles sont, en réalité, de tailles identiques. Afin de faciliter la comparaison, l’avant dernière voiture a été copiée au premier plan. *Image provenant de flickr sous licence Creative Commons de l'utilisateur pjfnsc27.*

Tailles relatives La taille relative est la mesure de la taille angulaire de la projection rétinienne d’au moins deux objets (ou textures) physiquement similaires [189, 25]. Les différences (ou équivalences) de tailles perçues d’objets que l’observateur sait identiques permettent d’identifier l’ordre dans lequel ces objets sont situés le long de l’axe optique. Ce mécanisme produit une information de nature ordinale mais permet dans certains cas d’estimer des ratios. Par exemple, un objet apparaissant deux fois plus grand qu’un autre est certainement deux fois plus proche. Il est illustré par la figure 2.7.

Densités relatives La densité relative correspond à la densité de la projection rétinienne d’un ensemble d’objets (ou de textures) qui sont placés de manière régulière dans le champ de vision [58]. Pour une texture, son grain apparaît de manière plus nette lorsqu’elle est proche de l’observateur. Pour des objets, ils apparaissent plus proches les uns des autres lorsqu’ils sont éloignés de l’observateur. Conceptuellement proche des tailles relatives, cet indice offre également une information de nature ordinale. Mais il permet aussi dans certain cas d’inférer des ratios. Par exemple, un ensemble d’objet apparaissant deux fois plus dense qu’un autre est certainement deux fois plus éloigné. Il est illustré sur la figure 2.8

Perspective atmosphérique La perspective atmosphérique fait référence à la difficulté croissante que l’on a à distinguer des objets lointains. Elle est principalement due à l’humidité et à la présence de polluants dans l’air qui entraînent une diminution du contraste entre des objets très éloignés et l’atmosphère : leur couleur se fond petit à petit avec la couleur de l’atmosphère. Cet indice offre une information de



FIGURE 2.8 – Illustration des densités relatives. Sur cette image de champ de riz, le grain est net au premier plan : on distingue parfaitement les différentes tiges. Cependant cet effet s’estompe avec la distance : les tiges sont plus nombreuses et apparaissent plus proches. *Image provenant de flickr sous licence Creative Commons de l'utilisateur Tadamasa Sawada.*



FIGURE 2.9 – Illustration de la perspective atmosphérique. Sur ce paysage, les montagnes de plus en plus éloignées se fondent petit à petit avec l’atmosphère, jusqu’à devenir difficilement discernables. *Image provenant de flickr sous licence Creative Commons de l'utilisateur Julius.Hibbert.*

nature ordinale et est illustré par la figure 2.9.

2.1.1.5 Autres mécanismes

Il semble qu’il y ait consensus dans la littérature pour considérer les mécanismes cités ci-dessus comme étant les uniques sources d’information de profondeur. D’autres sont parfois considérés, mais nous les avons omis dans cette liste pour les raisons suivantes.

La perspective linéaire est souvent considérée comme étant un mécanisme à part entière produisant un fort ressenti de profondeur [121] ; cependant elle semble résulter d’une combinaison de l’occlusion, des tailles relatives et des hauteurs dans le champ de vision [49].

La profondeur cinétique est mise en œuvre lorsqu’un objet est en rotation sur lui-

même. Bien que certains auteurs [124] la considèrent comme un mécanisme direct de perception de la profondeur, des études montrent qu’il s’agit plutôt d’une source d’information portant sur la forme de l’objet [30]. On retrouve cette discussion pour la luminance et les ombres qui apportent plus d’information sur la forme des objets que sur leur profondeur [149], exception faite des ombres portées [90].

Enfin, l’apparition et la disparition de textures se produisent quand un mouvement fait apparaître (resp. disparaître) tout ou partie d’un objet derrière un autre. Ce phénomène n’est en fait qu’un cas particulier de l’occlusion se produisant dans un contexte dynamique.

2.1.2 Importance relative des mécanismes de perception de la profondeur

Comme nous l’avons souligné précédemment, l’importance relative des mécanismes de perception de la profondeur dépend de la distance entre l’observateur et l’objet observé. Nous reprenons, sur la figure 2.10 un diagramme produit par Cutting [49] illustrant la relation entre la distance séparant des objets, et le “seuil différentiel ordinal”. Ce seuil est calculé à partir de la mesure de “seuil différentiel” [66] qui définit le seuil en dessous duquel deux *stimuli* ne sont plus différenciables. Lorsque les *stimuli* représentent des objets situés à différentes distances d’un observateur, il s’agit de déterminer l’ordre dans lequel ils sont placés dans l’axe optique. En considérant d_1 et d_2 les distances (en mètres) entre 2 objets et un observateur et avec d_1 l’objet le plus éloigné, ce seuil est calculé par le ratio entre la différence des distances de ces deux objets $d_1 - d_2$ et la moyenne des distances $(d_1 + d_2)/2$ soit : $2(d_1 - d_2)/(d_1 + d_2)$. Cette mesure est définie sur $[0, 1]$. Plus ce seuil est petit, plus les objets peuvent être proches l’un de l’autre et éloignés de l’observateur sans que cela n’affecte sa capacité à différencier le plus proche du plus éloigné.

La figure 2.10 indique que les différents mécanismes ne se comportent pas de la même façon avec la distance entre l’observateur et les objets observés. Nous remarquons que 3 mécanismes picturaux (occlusion, tailles relatives et densités relatives) n’y sont pas sensibles : leurs seuils respectifs sont constants. L’occlusion est d’ailleurs le mécanisme de perception offrant le seuil le plus performant permettant de différencier deux objets proches situés à une grande distance de l’observateur. La majorité des mécanismes sont cependant très sensibles à la distance et leurs seuils augmentent en fonction de celle-ci. Notons toutefois que le parallaxe (noté “Motion Perspective” sur le diagramme) a des performances qui s’améliorent avec la distance entre 1 et 3m. La perspective atmosphérique (notée “Aerial Perspective” sur le diagramme) a également une efficacité qui croît jusqu’à 2km avant de décroître.

En sus, pour tous les intervalles de distances considérés (espace personnel, espace d’action et espace de vue) plusieurs mécanismes se combinent. Des études

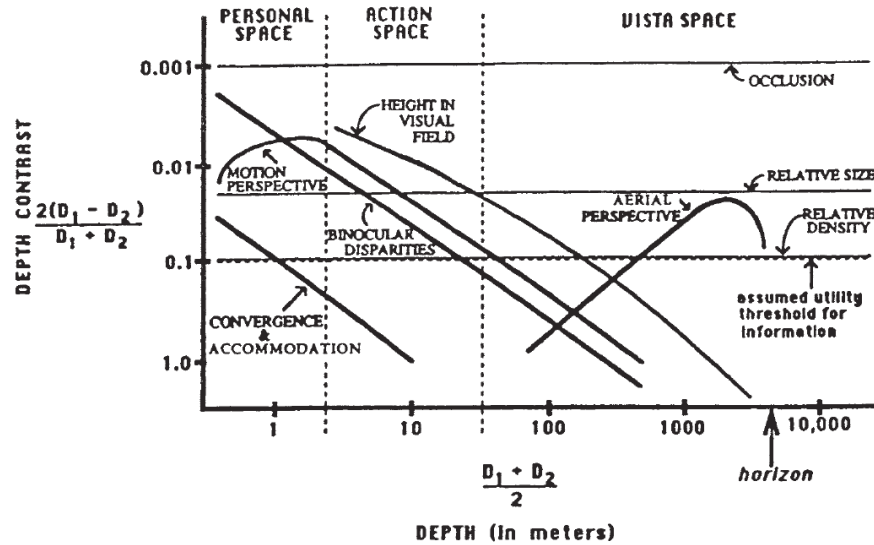


FIGURE 2.10 – Seuil différentiel ordinal en fonction de la distance entre l’observateur et les objets observés. Nous avons en ordonnée, une échelle logarithmique de la distance moyenne séparant l’observateur des objets ; et en abscisse, le seuil différentiel ordinal. Quelle que soit la distance, l’occlusion est toujours le mécanisme le plus important. Pour les distances les plus proches, la stéréoscopie et le parallaxe jouent également un rôle important ; tandis que pour les plus grandes distances, les densités relatives et la perspective atmosphérique rentrent en ligne de compte. Figure provenant de [49].

récentes [55, 201, 103] montrent qu’un modèle agrégeant les informations provenant de différents mécanismes est cohérent avec des observations expérimentales. Cependant, certains travaux [176] semblent suggérer que lorsqu’il y a suffisamment d’informations disponibles par le biais d’un ensemble restreint de mécanismes, d’autres peuvent être non appliqués. Par exemple, dans une tâche facile de parcours de labyrinthe 3D, l’ajout de la disparité binoculaire au parallaxe et l’occlusion n’apporte aucune amélioration de performance. *A contrario* l’ajout de la disparité binoculaire à l’occlusion (absence de parallaxe) conduit à une amélioration des performances.

2.1.3 Discussion

Dans ce chapitre, nous avons vu que la perception de la profondeur dépendait de 9 mécanismes qui interviennent comme des sources d’information pour le système visuel. Complémentaires, ils sont d’une précision variable en fonction de la distance séparant les objets de l’observateur.

Dans un système de visualisation visant à tirer parti de la perception de la profondeur, il est donc primordial de les intégrer dans la restitution graphique. D’après les recommandations de Cutting [49], la règle générale à suivre est d’employer les mécanismes ayant un seuil différentiel ordinal supérieur à 0.1. L’ajout systématique

de l'intégralité de ces mécanismes n'est cependant pas toujours pertinent. Et pour sélectionner les plus judicieux, en sus de l'occlusion qui est généralement bénéfique, plusieurs facteurs sont à prendre en compte.

Le premier facteur est la distance effective que le système doit reproduire. Ainsi, mis à part dans des simulateurs de vol [155] il est rarement nécessaire de simuler un effet de perspective atmosphérique. Celle-ci n'est en effet utile qu'à partir de quelques centaines de mètres, dans ce que Cutting appelle l'espace "vista" (*vista space*, au delà de 30m). Pour "l'espace personnel" (de 0 à 3m) et "l'espace d'action" (de 3 à 30m), tous les autres mécanismes sont éligibles.

Le deuxième facteur est le réalisme requis des objets à visualiser. En effet, l'observateur ne peut évaluer les rapports entre les tailles et densités relatives d'objets que si il est au préalable capable de les reconnaître. Dans le cas de la visualisation d'objets abstraits (en visualisation d'information, on parle alors de métaphores visuelles) ignorer ces mécanismes peut toutefois fausser la perception. Par exemple dans un système de visualisation de graphes où les nœuds sont représentés sous forme de sphères, et où aucun mécanisme n'est mis en œuvre, un observateur peut avoir du mal à discerner la différence entre une petite sphère située à une courte distance et une grande sphère plus éloignée.

Le troisième facteur est le réalisme de l'environnement dans lequel sont affichés les objets. Les hauteurs dans le champ de vision ne s'appliquent que si la gravité régit le placement de ces objets et si un sol plat est représenté. Outre ces facteurs, d'autres contraintes techniques ou matérielles, peuvent entrer en ligne de compte. Le parallaxe ne s'applique que si il est possible d'animer la restitution d'une façon ou d'une autre tandis que la disparité binoculaire requiert un système d'affichage permettant de produire une image différente pour chaque œil.

2.2 Apports et limites de la stéréoscopie

Dans cette section, nous nous focalisons sur la vision stéréoscopique. Et, sans prétendre à l'exhaustivité nous tentons d'établir un état des lieux des principaux apports de la stéréoscopie dans des systèmes de visualisation. Comme en rend compte le titre de l'article de Cockburn et McKenzie [41] "3D or not 3D ?" le débat sur la valeur ajoutée de la 3D en visualisation de l'information est vif et encore largement ouvert dans la communauté. Mais il repose en grande partie sur des rendus monoscopiques. Nous essayons d'alimenter le débat en tentant de mieux saisir les arguments favorables et défavorables à la stéréoscopie.

2.2.1 Etudes favorables à la stéréoscopie

Lorsque l'on compare la 3D stéréoscopique à la 3D monoscopique plusieurs propriétés sont généralement énoncées : (1) elle permet un meilleur jugement des profondeurs relatives ; (2) elle facilite la visualisation d'éléments spécifiques à travers un désordre visuel important [136] en facilitant la concentration sur des objets à différents niveaux de profondeur ; (3) elle permet de repérer des objets camouflés [199] ; (4) elle facilite l'appréhension de surfaces courbées et de leurs textures [98] ; et elle peut limiter l'impact et le désagrément produits par certaines dégradations (faible résolution, niveaux de gris limités, bruit..) [136].

Ces propriétés ont été éprouvées et complétées dans différentes études expérimentales en particulier dans le domaine médical. En effet, la médecine est certainement un des domaines pionniers de la visualisation stéréoscopique. Depuis le *XIX^e* siècle et les travaux de Röntgen sur les rayons X l'imagerie médicale est devenue un outil incontournable de diagnostic. D'abord imprimées, les images résultantes des différentes techniques d'imagerie sont désormais consultables via des systèmes de visualisation spécialisés [170]. L'avènement de la tomographie (reconstruction d'un volume à partir d'une série de coupes) a également permis le développement de techniques de restitution 3D de ces données. Les données originales étant naturellement volumiques (organes, os, etc...), il est légitime de se demander si la stéréoscopie peut aider aux différentes tâches liées à la visualisation médicale.

En préambule, il est important de distinguer le “rendu géométrique” (*Surface Rendering*) du “rendu volumique” (*Volumetric Rendering*) qui sont deux techniques qui permettent de générer une restitution tridimensionnelle à partir de données volumiques. Ces types de données sont classiques en imagerie médicale (échographe, IRM, scanner etc...) et se présentent sous la forme d'une série d'images de coupes de l'objet à visualiser. Le rendu géométrique (voir figure 2.11) consiste à extraire une approximation polygonale de la structure à visualiser via des algorithmes comme le *Marching Cubes* [130] ou le *Marching Tetrahedra* [54]. Le rendu volumique (voir figure 2.12), consiste à combiner toutes les coupes alignées. Plusieurs approches peuvent être employées comme par exemple l'accumulation sur un rayon [132] ou un opérateur de mélange [117]. Les différentes couches se trouvent alors le plus souvent discernables par transparence.

A notre connaissance, Rosenbaum et al. [167] sont parmi les premiers à avoir étudié l'impact de la stéréoscopie dans un système de visualisation d'imagerie médicale de rendu géométrique 3D provenant de différentes sources (tomodensitométrie, résonance magnétique et angiographie). Dans une tâche d'identification de structures anatomiques dans des images de squelettes, ils trouvent que la stéréo-

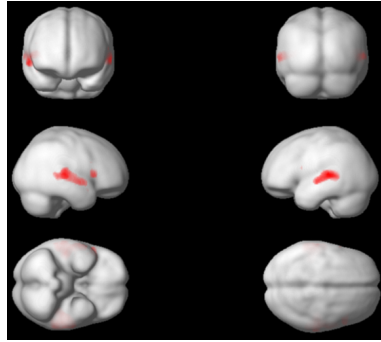


FIGURE 2.11 – Exemple de rendu géométrique. Seule l’enveloppe extérieure du volume est visible.

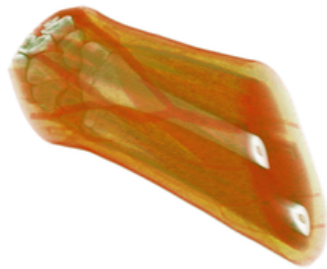


FIGURE 2.12 – Exemple de rendu volumique. Toutes les couches du volume sont visibles par transparence, mais les surfaces apparaissent moins bien.

scopie obtient un taux d’identification et de confiance plus élevé que la monoscopie. Cependant, d’après leur protocole expérimental, chaque participant effectuait la même tâche sous, successivement, une condition monoscopique puis une condition stéréoscopique. Leurs résultats peuvent donc être partiellement dûs à un biais d’apprentissage.

En visualisation de tissus mous l’imagerie ultrasonique est très utilisée. Rapide et non-intrusive, cette technique a cependant l’inconvénient de produire des images très bruitées. L’utilisation de la stéréoscopie pour la visualisation de ce type d’images semble être bénéfique d’après [97] qui rapporte que cette technique facilite le diagnostic du cancer du sein. Ces résultats sont confirmés, pour une même tâche, et dans des conditions d’acquisition similaires (imagerie par rayons X) par [85, 101]. Le gain apporté par la stéréoscopie semble venir du bruit contenu naturellement dans ces images, qui impacte moins l’utilisateur en 3D stéréoscopique. Dans une tâche différente, il a été observé [115] que les participants identifient plus facilement le sens de rotation d’un *stimulus* visuel sphérique dans une image très bruitée émulant une reconstruction radiographique digitale.

Plus récemment, Cho et al. se sont penchés sur le problème de la visualisation de vaisseaux sanguins par rendu volumique qui se caractérise par des restitutions avec

beaucoup de transparence, d'occlusion et des structures spatiales ambiguës [40]. Leurs travaux se basent sur un jeu de données artificielles représentant des images bruitées de croisements de tubes simulant le croisement des vaisseaux sanguins. Ils demandent aux participants de déterminer quels sont les tubes qui se situent au-dessus ou en dessous des autres. Leurs résultats indiquent que la stéréoscopie améliore les performances pour cette tâche. Hormis l'occlusion, aucun mécanisme de perception ne peut être implémenté directement à partir de ces images ; la stéréoscopie est donc le principal indice de profondeur et facilite ainsi l'identification des distances relatives entre les objets visualisés. Nelson et al. [145] confirment que la stéréoscopie améliore la détection de structures chevauchantes dans une tâche d'identification de structures anatomiques dans d'autres rendus volumiques obtenues par imagerie ultrasonique.

L'usage de la stéréoscopie a aussi été extensivement étudié dans le secteur de la robotique, pour notamment les téléopérations où un opérateur effectue une action à distance via un appareil robotisé. Le plus souvent, ces appareils sont équipés de dispositifs d'acquisition vidéo permettant un retour visuel à l'opérateur ; on parle alors de télé-présence. De nombreuses études relatent les apports de la stéréoscopie dans ce cadre [43, 21, 79]. Cole et al. affirment par exemple que certaines tâches sont impossibles sans un retour stéréoscopique, comme par exemple la manipulation à distance d'une tige à travers un labyrinthe de fils de fer. Dans ce type de situation, la stéréoscopie permet de rendre le flux vidéo plus proche des sensations *in situ* via la perception de la profondeur, ce qui facilite l'exécution de ces tâches demandant une grande dextérité.

En réalité augmentée, où des images virtuelles sont superposées à des images réelles, l'utilisation de la stéréoscopie améliore la performance des utilisateurs dans une tâche d'identification d'étiquettes virtuellement associées à des objets réels [160, 159]. L'affichage de ces étiquettes comporte de nombreux recouvrements et rend la tâche difficile quand aucun algorithme d'optimisation de placement n'est employé. Tant que ces étiquettes sont affichées avec une disparité correspondant à la distance à laquelle l'objet étiqueté se trouve, la stéréoscopie semble plus efficace que la monoscopie. Effectivement, elle permet de mieux distinguer des objets se trouvant à des profondeurs différentes, tout en rendant possible la visualisation à travers des amas. Nous pouvons mettre ces travaux en parallèle à ceux de [33] qui utilisent la stéréoscopie afin de surligner certains composants d'une interface 2D d'un ordinateur de bord destiné aux voitures (GPS, navigation dans des menus, musique etc..). La stéréoscopie permet ainsi de distinguer plus rapidement et plus efficacement les éléments sélectionnés du reste puisqu'ils apparaissent à une distance plus proche de l'utilisateur. Les auteurs affirment également que l'ajout de la

stéréoscopie ne semble entraîner aucune surcharge cognitive par rapport à la condition initiale. Dans leurs expérimentations, les utilisateurs ont préféré cette solution. Il est d'ailleurs important de noter que, globalement, la stéréoscopie est attrayante et est souvent préférée par les utilisateurs, en particulier dans les jeux vidéos [175].

2.2.2 Etudes défavorables à la stéréoscopie

Nous avons présenté dans la partie précédente, un certain nombre de travaux rapportant une valeur ajoutée de la stéréoscopie dans des applications variées. Dans cette partie, nous relatons celles qui rapportent un impact négligeable ou des pertes de performances eu égard à la monoscopie.

En visualisation médicale, les études consacrées aux restitutions volumiques concluent généralement sur un apport positif de la stéréoscopie. Les résultats correspondant aux restitutions obtenues par rendu géométrique sont plus nuancées. Par exemple, dans une étude d'impact de la stéréo et de la taille de l'écran pour l'analyse d'images d'IRMs, Chen et al. [38] montrent que, contrairement à leurs hypothèses initiales, la taille de l'écran n'a pas d'impact sur le résultat de l'analyse (en terme de temps de réponse et d'erreur dans différentes tâches) et que l'ajout de la stéréoscopie a un effet négatif. Cependant, certains choix expérimentaux semblent discutables. Le protocole expérimental emploie deux écrans de tailles différentes (un petit et un grand) situés à différentes distances de façon à ce que la projection rétinienne soit identique. Dans les deux cas, la projection stéréoscopique est identique alors qu'elle devrait être recalibrée, puisqu'elle dépend de la distance entre l'observateur et l'objet observé. De plus, le contraste, caractérisé comme un élément primordial de la visualisation par les auteurs, n'est pas constant entre les conditions. Sous la condition monoscopique, les utilisateurs ne gardent pas les lunettes à obturation, et les images perçues sous la condition stéréoscopique sont donc 50% plus sombres.

Une autre étude portant sur un rendu géométrique pour une tâche d'analyse de fractures de l'acetabulum (articulation au niveau de la hanche) conclut à l'absence de différence significative entre les tomographies visualisées en monoscopie et en stéréoscopie [116]. Des conclusions similaires sont rendues pour une tâche de diagnostic de cancer du poumon par visualisation de tomographies [198]. Les auteurs observent cependant une tendance vers une meilleure détection des nodules en stéréoscopie qu'en monoscopie ou avec des vues en coupes 2D.

Dans une tâche d'analyse de labyrinthe en 3D [176] (imitant l'étude de structures vasculaires), les auteurs rapportent que la stéréoscopie n'apporte aucun bénéfice -en temps de réponse- lorsque le parallaxe est déjà présent mais qu'elle est bénéfique lorsqu'il est absent. Cependant, les auteurs notent la simplicité de la tâche et la pré-

sence d'autres indices de profondeurs dans la visualisation (occlusion, ombrage). L'impact de la stéréoscopie sur le nombre d'erreurs commises n'est pas mesurée dans cette étude.

Drascic et al. ont étudié, dans le cadre d'une application de téléopération [57], les effets d'apprentissage entre des restitutions monoscopique et stéréoscopique. Pour une tâche simple (déplacer un appareil robotisé d'un point A vers un point B) et une tâche plus compliquée (déplacer l'appareil et positionner un bras robotisé en évitant des obstacles), ils observent qu'au début de la tâche, la condition stéréoscopie recueille les meilleurs résultats, mais que cet effet s'estompe avec les répétitions (8 répétitions), jusqu'à l'obtention de résultats équivalents pour les deux conditions. Toutefois, [72] considère qu'une tâche permettant un tel apprentissage n'est pas commune dans le domaine de la téléopération.

2.2.3 Discussion

L'apport de la stéréoscopie dépend, comme toute technique de fouille (visuelle ou non), de la nature des données et de la tâche concernée. Il ressort de notre analyse des travaux de la littérature, que la stéréoscopie semble bénéfique dans les cas suivants : données photo réalistes, tâches nécessitant une précision particulière, données intrinsèquement 3D ne disposant pas ou peu d'autres indices de profondeurs ou ayant des indices contradictoires (ou pas suffisamment discriminants) ; données complexes ou tâches complexes (ce qui contrebalance l'adaptation nécessaire à la stéréo, partiellement confirmée par les résultats de [23]).

En visualisation d'images médicales [22], la stéréoscopie semble bénéfique dans la visualisation de données 3D volumiques. Son apport reste cependant mitigé concernant les restitutions géométriques [116, 167]. Il faut néanmoins noter, que dans les études considérées ici, les tâches effectuées avec la stéréoscopie n'ont jamais de résultats inférieurs à la condition expérimentale originale [22].

2.3 La visualisation immersive de graphes

Dans cette section, nous proposons un état des lieux des études relatant de la visualisation immersive de graphes. Nous nous concentrons dans un premier temps sur les systèmes stéréoscopiques avant de présenter brièvement des systèmes avec une composante immersive plus importante.

2.3.1 La visualisation 3D stéréoscopique de graphes

Dans la communauté *Graph Drawing*, de nombreux systèmes proposant une restitution 3D monoscopique ont été proposés avec un essor marqué au début des années 90 [28, 104, 10, 13, 195, 48, 84, 51, 153]. La majorité d’entre eux concernent les tracés orthogonaux tri-dimensionnels [152, 65] où la position des nœuds et des liens est contraint à une grille. D’autres travaux ont exploré des tracés dans des espaces hyperboliques [178, 141] où la position des nœuds et des liens sont contraints à des sphères concentriques.

A notre connaissance, une des premières études analysant le potentiel de restitutions en 3D stéréoscopique dans la visualisation de graphes a été menée par Ware et Franck en 1996 [200]. Elle montre pour différentes expériences que cette méthode de restitution permet aux utilisateurs d’interpréter de plus grands graphes qu’en 2D (environ 60% plus grands). La tâche considérée était une tâche d’identification de chaînes entre nœuds. Il s’agissait de déterminer s’il existe une chaîne de taille 2 entre 2 nœuds surlignés. Dans une étude antérieure, Sollenberger et Millgram [185] avaient montré que dans une tâche similaire d’identification de chaînes dans un labyrinthe, le parallaxe (implémenté par une rotation autour des objets visualisés) est plus important que la disparité binoculaire.

Plus d’une décennie après la publication de ces études, la qualité des appareils de restitution stéréoscopique s’est considérablement accrue. Pour évaluer l’impact de cette évolution, Ware et Mitchell [202] ont reproduit leur expérience avec du matériel plus récent. Leurs résultats indiquent que les gains de performance obtenus grâce au parallaxe et la disparité binoculaire sont additifs et qu’ils permettent aux utilisateurs de résoudre la tâche sur des graphes ayant jusqu’à 1000 nœuds avec moins de 10% de taux d’erreur. Ceci représente une amélioration d’un ordre de magnitude par rapport à la 2D.

Plus récemment, Alper et al. [3] ont étudié une combinaison de la stéréoscopie avec une restitution 2D pour surligner certains nœuds d’un graphe (appelée 2.5D). Initialement, tous les nœuds sont représentés sur un plan. Lorsqu’un ensemble de nœuds est sélectionné, il est affiché dans un plan plus proche de l’utilisateur. Pour différentes tâches locales (identification de voisins ou de voisins mutuels), la différence d’efficacité et de temps de réponse avec une technique plus classique de surlignage coloré n’est pas significative. Cependant, la combinaison de ces deux techniques (2.5DH) obtient les meilleures performances pour les temps de réponse et les taux d’erreur. Cette combinaison a été comparée à une restitution en “3D stéréoscopique complète” où tout le graphe est affiché en 3D stéréoscopique et où le surlignage est coloré. Les résultats sont mitigés : il n’y a aucune différence significative entre la 3D stéréoscopique et la 2.5DH pour les erreurs, hormis pour pour une

tâche locale de comptage de nœuds surlignés pour laquelle la 2.5DH entraîne moins d'erreurs. Globalement, les temps de réponses sont plus faibles pour la 2.5DH. Il est cependant important de noter que cette expérience n'intègre aucune interaction ou rotation permettant de ressentir le parallaxe, qui, conjointement à la stéréoscopie, permet un ressenti de la profondeur plus fidèle.

Les travaux précédents ont été menés avec des périphériques d'affichages assez classiques *e.g.* un écran à haute fréquence de rafraîchissement avec lunettes stéréoscopiques à obturation, ou un stéréoscope de Wheatstone (arrangement de miroirs reflétant les images de deux écrans). Il existe cependant d'autres études basées sur des affichages moins standards construits spécifiquement pour tenter de mesurer l'impact de la stéréoscopie. Par exemple, [19] propose un système de réalité augmentée pour la visualisation de graphes complexes en 3D, dans lequel les graphes sont affichés au dessus d'un morceau de carton servant de socle permettant de manipuler le graphe. Pour la même tâche que celle étudiée par Ware et al., les auteurs montrent que le système de réalité augmentée d'une part, et une restitution 3D stéréoscopique d'autre part, se comportent mieux (en temps de réponse et nombre d'erreurs) qu'une restitution 2D. La combinaison de la disparité binoculaire avec le système de réalité augmentée améliore les performances.

2.3.2 Visualisation hautement immersive de graphes

L'émergence de systèmes hautement immersifs de type *Cave* a conduit à quelques expérimentations en visualisation de graphes [209, 92, 151]. Bien que ce dispositif sorte du cadre de ce document, il est toutefois intéressant de noter que les résultats obtenus sont prometteurs. Halpin et al. [92] proposent un système similaire à celui de Alpert et al. [3] en utilisant la stéréoscopie comme système de surlignage de nœuds ; l'utilisateur peut sélectionner un nœud via un périphérique d'interaction de type *Wand* (l'utilisateur manipule une baguette qui contrôle le mouvement d'un pointeur dans la restitution) et le "tirer" vers lui. Cette action permet de déplacer le nœud sélectionné et ses voisins vers un plan plus proche de l'utilisateur. L'évaluation de ce système semble confirmer les résultats précédents : la stéréoscopie améliore l'analyse fine du graphe. Les auteurs notent que cette approche, ne permettant de surligner qu'un nœud et ses voisins, n'aurait que peu d'intérêt pour une analyse plus globale. Yang et al. [209] présentent un prototype de système de visualisation de réseaux métaboliques qui est aussi basé sur l'utilisation d'une *Wand* au sein d'une *Cave*. Dans un système similaire, Osawa et al. [151] remplacent l'usage de la *Wand* avec un système de suivi permettant de connaître la position du pouce et de l'index de l'utilisateur pour déclencher les interactions. Ces deux études ne

rapportent aucune expérience utilisateur. D’autres prototypes sont présentés dans la littérature, comme ceux de Hérissou et al. [95] et Férey et al. [70]. Ils permettent de visualiser des données biologiques (chaînes nucléiques et protéines) qui sont spatialement structurées et partagent de nombreux points communs avec les graphes. A notre connaissance, Hérissou et al. n’ont pas conduit d’expérience utilisateur. En revanche, Férey et al. notent que leur prototype a été validé par des experts biologistes pour l’exploration d’un graphe représentant des gènes co-régulés où des nœuds sont liés si les gènes qui leur correspondent sont co-régulés. Le protocole employé n’est cependant pas décrit.

2.3.3 Discussion

Les expérimentations que nous avons relatées dans cette section portent toutes sur des tâches que nous qualifions de locales : identification d’un nœud et de ses voisins, identification de l’existence d’une chaîne entre nœuds, comptage de nœuds. Pour ces tâches, la comparaison de la stéréoscopie avec la 2D et la 3D monoscopique montre une valeur ajoutée de la 3D stéréoscopique en taux d’erreur et temps de réponse. La combinaison de la stéréoscopie avec la 2D semble aussi prometteuse. Cependant, les tâches requises en pratique en fouille visuelle de réseaux ne se limitent pas à ces tâches locales et nécessitent des analyses à l’échelle macroscopique pour extraire des propriétés structurelles globales (ex : communautés).

Fouille visuelle de graphes

Visual analytics is more than visualization and can rather be seen as an integrated approach combining visualization, human factors and data analysis.

—Daniel Keim

Sommaire

Introduction	42
3.1 Tâches en fouille de graphes	43
3.1.1 Typologie générale	44
3.1.2 Détection de communautés	46
3.2 Tracés de graphes	47
3.2.1 Modèles de représentation et algorithmes de tracé	47
3.2.2 Tracés de type force et ressort	49
3.2.3 Tracés 3D monoscopique	51
3.3 Interactions en fouille de graphes	52

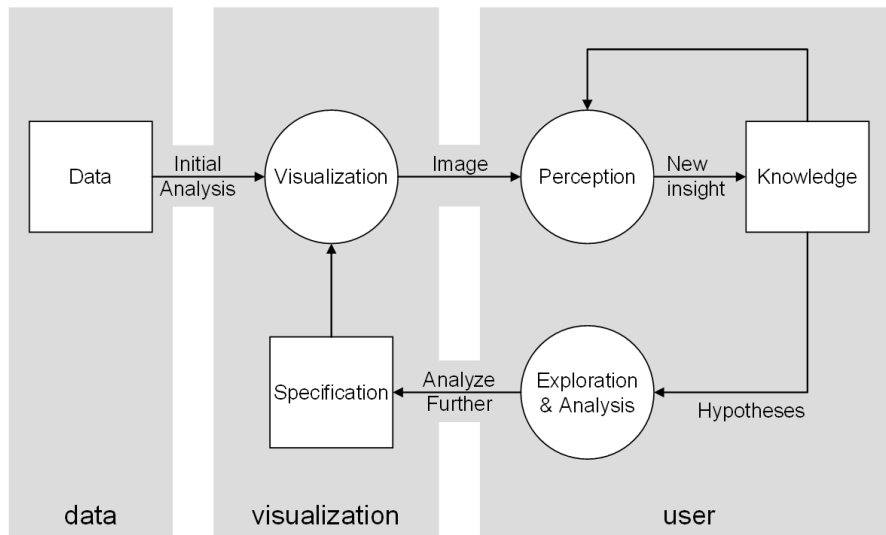


FIGURE 3.1 – Illustration du pipeline de fouille visuelle d’après [113] et issu du modèle de visualisation de [194]. Les cercles représentent des processus tandis que les rectangles représentent des conteneurs. À partir d’une analyse initiale des données, l’utilisateur rentre dans une boucle de raisonnement. Ce processus permet d’assimiler de nouvelles connaissances à partir des images restituées, ce qui peut entraîner la formulation de nouvelles hypothèses. Celles-ci peuvent ensuite être évaluées via des interactions ou des procédés d’analyse automatiques qui peuvent impacter la visualisation.

Introduction

Comme nous l’avons vu dans l’introduction de ce manuscrit, la fouille visuelle de graphes, et en particulier de réseaux sociaux, est devenue un enjeu majeur dans les communautés de visualisation de l’information et de fouille de données. Elle s’inscrit dans le cadre plus général de la fouille visuelle ou *Visual Analytics* que Thomas et Cook ont définie en 2005 comme étant “la science du raisonnement analytique facilité par des interfaces visuelles interactives” [191]. En 2008 et en s’appuyant sur le modèle de Wijk [194], Keim a modélisé le processus de fouille visuelle comme une boucle de raisonnement structurant la découverte d’informations (illustrée sur la figure 3.1). À partir d’une analyse initiale des données (par exemple l’extraction de la plus grande composante connexe d’un graphe), une représentation initiale peut être choisie. Le processus entre ensuite dans une boucle dans laquelle l’utilisateur assimile de nouvelles connaissances à partir des images qui lui sont restituées (processus de visualisation et de perception). Ces connaissances peuvent entraîner la formulation d’hypothèses qui sont ensuite évaluées en manipulant la représentation des données (processus d’exploration). Ces interactions entraînent la restitution de nouvelles images qui vont à leur tour permettre d’affiner les connaissances de l’utilisateur.

Par rapport à ce modèle, l’apport principal de cette thèse réside dans l’analyse

de l'impact qu'entraîne l'ajout de la stéréoscopie dans le processus de visualisation sur les processus de perception et d'exploration. Comme le souligne Keim [113], la fouille visuelle est une discipline orientée application et les besoins du domaine applicatif ainsi que les propriétés des données étudiées structurent le processus de fouille. Il est donc primordial de rappeler certains points de la fouille visuelle de graphes.

Sans remonter dans les années 60 aux travaux précurseurs de Knuth [118] et de Tutte [193] en dessin de graphes, un état de l'art exhaustif de ce domaine s'avère aujourd'hui bien délicat tant le volume de publications a explosé ces deux dernières décennies. Ainsi, sans prétendre aucunement à l'exhaustivité ce chapitre a pour objectif de présenter, en les positionnant par rapport aux grandes approches existantes et au processus de fouille visuelle, les choix de modèles de représentation, d'algorithmes de tracé et de tâches de fouille sur lesquels repose la suite de cette thèse.

Bien que l'identification des tâches d'un processus de fouille soit une question qui fasse encore largement débat, nous nous appuyons dans la première section sur les typologies de tâches de Lee et al. [126] et de Amar et al. [4] pour identifier les tâches majeures au regard de l'analyse des réseaux sociaux : les tâches orientées attributs, les tâches de navigation, les tâches d'estimation et les tâches orientées topologie. Nous nous focalisons ensuite sur les différents types de tâches de détection de communautés. Dans la deuxième section, nous rappelons les principaux modèles de restitution visuelle de graphes à savoir le modèle nœud-lien, les représentations matricielles et les *treemaps*. Puis nous présentons le problème de tracé de graphes ainsi que les techniques les plus populaires avant de nous concentrer sur les tracés de type force et ressort et sur les tracés 3D monoscopiques. Dans la troisième section, nous présentons les principales techniques d'interaction qui facilitent l'exploration et l'analyse des données via la manipulation de la restitution visuelle.

3.1 Tâches en fouille visuelle de graphes

En amont de la boucle de raisonnement de la fouille visuelle se trouve une première étape d'analyse des données. Outre l'application des techniques de prétraitement classiques (sélection de variables, échantillonnage etc..), cette étape permet de considérer deux éléments structurants de l'analyse : (i) le type de structures à extraire des données et (ii) le niveau de granularité auquel celles-ci vont être étudiées. En fouille de graphe par exemple, différents algorithmes d'extraction automatique ou de visualisation sont plus adaptés à la fouille de communautés tandis que d'autres permettent d'identifier certains individus clefs. Dans cette section, nous tentons d'établir une typologie des tâches les plus courantes en analyse de réseaux sociaux pour identifier ces structures.

3.1.1 Typologie générale

En fouille de données, et plus particulièrement en fouille de données exploratoire, il est commun d’être confronté à un jeu de données sans aucune hypothèse *a priori* sur les informations qu’il recèle. Il devient alors difficile d’évaluer la pertinence ou de comparer différentes approches puisque la marche à suivre pour découvrir les pépites de connaissance est inconnue. Dans un tel contexte, des typologies de tâches sont formulées par la communauté pour fournir certaines directives générales permettant d’aborder des problèmes de fouille complexes. En fouille visuelle de graphes, la typologie de tâche rédigée par Lee et al. en 2006 [126] fait office de référence et est largement utilisée (par exemple par [94, 184, 142]). Cette liste de tâches courantes s’appuie sur une typologie antérieure [4] qui explicite les tâches auxquelles la communauté de fouille visuelle se trouve le plus souvent confrontée. Une reproduction de cette typologie de fouille visuelle est proposée sur le tableau 3.1.

Tâche	Description
Récupérer une valeur	Connaissant un ensemble de cas, trouver leurs attributs
Filtrer	Trouver les cas satisfaisant certaines conditions portant sur leurs attributs
Calculer des valeurs dérivées	Connaissant un ensemble de cas, calculer une agrégation numérique (moyenne, médiane, compte)
Trouver les extrema	Trouver les cas possédant une valeur extrême pour l’un de ses attributs
Trier	Trier les cas selon une métrique ordinale
Gamme de valeurs	Selon un ensemble de cas, déterminer l’éventail des valeurs d’un attribut
Caractérisation de distribution	Selon un ensemble de cas et un attribut quantitatif, caractériser la distribution de cet attribut sur tout l’ensemble
Trouver les anomalies	Identifier les anomalies d’un ensemble de cas par rapport à certaines attentes (par exemple statistiques)
Agréger	A partir d’un ensemble de cas, agréger les cas ayant des attributs similaires
Corréler	A partir d’un ensemble de cas et de deux attributs, déterminer les relations utiles entre leurs valeurs.

TABLE 3.1: Typologie de tâches pour la fouille visuelle de données tabulaires proposée par [4].

En appliquant ces tâches de fouille visuelle à la fouille visuelle de graphes, Lee et al. parviennent à dresser la typologie suivante, dont les tâches sont réparties parmi quatre catégories : tâches orientées topologie, tâches orientées attributs, tâches de navigation et tâches d’estimation.

Tâches orientées attributs Cette catégorie comporte les tâches nécessitant d'analyser les valeurs associées aux nœuds ou au liens. Elle se décompose en deux sous-catégories qui sont : les tâches portant sur les nœuds et les tâches portant sur les liens. Il s'agit alors par exemple de trouver la liste des nœuds ayant une valeur d'attribut spécifique ou encore, pour un nœud particulier, de trouver ses voisins connectés par un certain type de lien.

Tâches de navigation Dans cette catégorie, Lee et al. regroupent les tâches demandant de "naviguer" dans le graphe en passant d'un nœud à l'autre en suivant un chemin particulier (ou vérifiant certaines propriétés) tout en le mémorisant ; ce qui permet des retours en arrière.

Tâches d'estimation Elles demandent de pouvoir estimer une valeur rapidement. Lee et al. citent par exemple une tâche visant à estimer la taille d'un réseau social.

Tâches orientées topologie Cette catégorie comporte les tâches permettant d'identifier des structures en se basant sur la structure topologique d'un graphe. Par ordre de granularité croissant, nous pouvons ainsi lister les sous-catégories suivantes :

- Adjacence : ces tâches de bas niveau prennent uniquement en compte un nœud et ses voisins. Généralement il s'agit donc de trouver l'ensemble des voisins d'un nœud ou de les compter.
- Accessibilité : d'un niveau de granularité plus élevé, il s'agit de trouver les nœuds qui sont accessibles à partir d'un nœud d'origine. Ce procédé peut être considéré comme une répétition de tâches d'adjacence.
- Connexions communes : il s'agit de trouver des nœuds qui sont connectés de la même façon à un autre ensemble de nœuds.
- Connectivité : cette sous-catégorie décrit les tâches portant sur la connectivité dans un sous-graphe. Nous pouvons par exemple citer : l'identification de communautés, l'identification de composantes connexes et l'identification de points d'articulation.

A ces quatre catégories, ils ajoutent des tâches de plus haut niveau comme par exemple la comparaison de deux graphes, l'analyse de l'évolution d'un graphe dans le temps, ou l'identification de doublons dans la représentation. Ces tâches font intervenir des connaissances exogènes supplémentaires, comme un autre graphe pour la comparaison, ou des propriétés sémantiques (ex : un nœud en double est incorrect).

3.1.2 Détection de communautés

Dans cette thèse, nous nous concentrons sur une tâche orientée topologie dépendante de la connectivité à l'intérieur des graphes : la détection de communautés. Notons que pour un graphe modélisant un réseau social, la notion de communauté regroupant des individus est intuitive et pertinente, mais dans le cas de graphes représentant d'autres types de données on parle plus souvent de "structures en communautés" ou de classes et de partitions (*clusters*). Girvan et Newman, qui ont notablement contribué à la détection automatique de communautés, les définissent comme : "*the property of community structure, in which network nodes are joined together in tightly knit groups, between which there are only looser connections*" [88]. Depuis, la notion de structures en communautés a évolué vers une recherche de typologie de communautés. Dans leur livre, Tang et al. [188] font ainsi la distinction entre différents types de communautés qui vont diriger l'exercice de détection. Ceux-ci ne sont pas mutuellement exclusifs.

- "*Node-centric community*" : chacun des nœuds d'une communauté remplit certaines conditions/propriétés (ex. : former une clique).
- "*Group-centric community*" : ne considère que les connexions à l'intérieur d'un groupe qui remplit certaines propriétés sans zoomer vers une analyse plus fine (au niveau des nœuds), un groupe ayant par exemple une certaine densité.
- "*Hierarchy-centric community*" : construction d'une structure hiérarchique de communautés. Les structures en communautés de ce type se retrouvent souvent dans des grands graphes. A l'échelle d'une région, on peut par exemple retrouver une structure hiérarchique représentant les quartiers, puis les villes et les départements.
- "*Network-centric community*" : partitionnement de l'intégralité du réseau en des ensembles disjoints de nœuds.

De très nombreux algorithmes de détection de communautés ont été proposés dans la littérature pour traiter ces différents types de communautés [74]. Un état de l'art de ces approches sort du cadre de cette thèse, mais nous pouvons noter qu'elles peuvent être catégorisées selon trois grands axes principaux : les approches de détection de communautés sans recouvrement (un nœud ne peut appartenir qu'à une seule communauté), avec recouvrement, et les approches de détection de communautés hiérarchiques. La littérature se concentre principalement sur des détections dans des réseaux statiques mais des travaux récents abordent les problématiques de détection de communautés dans des réseaux dynamiques (e.g. [36]).

Dans la suite de ce manuscrit, nous nous focalisons sur une tâche de détection de communautés orientée réseau : un partitionnement du réseau en un ensemble de communautés disjointes.

3.2 Tracés de graphes

Dans la boucle de raisonnement du processus de fouille visuelle, l'objectif est de trouver une représentation des données adaptée qui permette de faciliter la perception des structures à découvrir. Comme nous l'avons mentionné plus haut, il existe une multitude de représentations de graphes développées pour répondre à des problématiques différentes. Dans cette section, nous rappelons les principaux modèles de représentation historiques et nous nous concentrons ensuite sur le modèle nœud-lien et sur les tracés de type force et ressort.

3.2.1 Modèles de représentation et algorithmes de tracé

D'après une analyse menée par L.C. Freeman [75], les modèles de visualisation les plus utilisés se classent en deux grandes familles. Le premier paradigme, le plus proche du modèle théorique des graphes, est le modèle nœud-lien dans lequel les acteurs sont représentés par des nœuds et les connexions sociales par des arêtes. Le second paradigme repose sur l'utilisation de matrices dont les colonnes et lignes représentent les acteurs, et les cellules les connexions. A ces deux familles, nous pouvons également ajouter les treemaps de Shneiderman [182, 180]. Initialement développée pour visualiser le contenu d'un disque dur, cette approche a été récemment étendue à la visualisation de graphes [68] et nécessite de décomposer le graphe en un arbre couvrant et le reste des liens. L'arbre couvrant est alors affiché sous forme de treemap sur laquelle les liens restants sont superposés. Mais si les données ne sont pas fortement structurées de manière hiérarchique, la superposition de ces liens entraîne vite une complexité visuelle importante.

Le modèle nœud-lien a été de loin le plus utilisé, probablement en raison de sa proximité avec les outils combinatoires utilisés dans l'analyse [71]. Notons cependant que les comparaisons expérimentales de ce modèle avec le modèle matriciel dans un cadre général ont montré la supériorité de ce dernier dans la résolution de tâches de "bas niveau" (comme trouver un voisin commun ou un plus court chemin entre deux paires de nœuds) sur des réseaux de quelques dizaines de nœuds. [86]. Ces résultats ont été confirmés récemment dans l'analyse des réseaux sociaux [94]. Cependant, les praticiens savent bien que l'étude d'un réseau social ne se réduit pas à une succession de tâches de bas niveau. Des analyses complémentaires effectuées par les auteurs pré-cités ont montré que pour la détection de certaines structures (ex : identification d'un point d'articulation), le modèle nœud-lien était le plus performant. Ainsi, il semble que les restitutions matricielles soient bien adaptées à la mise en évidence de tendances globales qui constitue la première phase du mantra de la visualisation de l'information "Overview first, zoom and filter then detail on demand" [181]. En revanche, elles montrent leurs limites dans une analyse plus fine

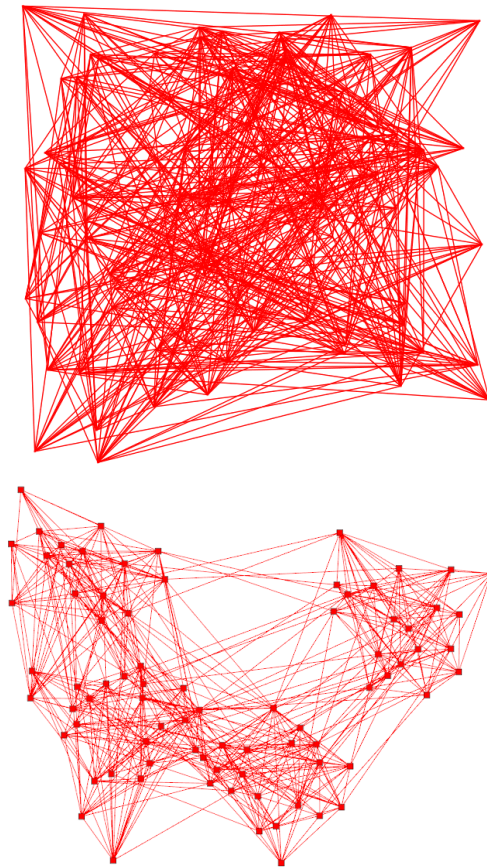


FIGURE 3.2 – Illustration de deux tracés d’un même graphe. La première capture représente un tracé de type aléatoire tandis que la seconde représente un tracé de type force et ressort. Dans les deux cas, le graphe représenté possède une structure en communautés comme celles que nous allons considérer dans le chapitre 5, de paramètres $k = 4$, $nv = 20$, $p_{int} = 0.6$, $p_{ext} = 0.05$. Les 4 communautés composant ce graphe n’apparaissent pas avec le tracé aléatoire tandis qu’elles apparaissent avec le tracé force et ressort. *Captures d’écran du logiciel Tulip [9].*

de la structure topologique. Ainsi, nous nous focaliserons dans la suite sur le modèle nœud-lien.

Le bon déroulement du processus de fouille visuelle d’un graphe représenté par le modèle nœud-lien dépend en grande partie du choix de l’algorithme de tracé utilisé. Ces algorithmes permettent de placer les nœuds et liens dans l’espace et le tracé final varie grandement en fonction du type d’algorithme utilisé (une illustration est proposée sur la figure 3.2).

Le but *in fine* de tout algorithme de tracé de graphes est de présenter une restitution graphique d’un graphe qui représente précisément et efficacement l’information “utile” qu’il contient. Dans la communauté de visualisation de graphes, plusieurs critères esthétiques sont souvent mentionnés afin de caractériser ces tracés, comme par exemple des liens rectilignes ou des croisements limités (se référer à [203] pour une revue des principaux critères). Dans ce contexte, un problème générique de

tracé de graphes peut se poser comme un problème d’optimisation combinatoire multi-objectif qui ne tient pas compte de la sémantique des objets manipulés : plonger les sommets dans un espace (généralement un plan) euclidien en minimisant le nombre de croisement d’arêtes et en conservant des angles supérieurs à un seuil entre les segments de droite représentant les arêtes [52]. La situation peut être très différente lorsque les structures topologiques sont associées explicitement à des informations [96]. Historiquement, les premiers critères esthétiques étaient d’ailleurs souvent contraints par la sémantique des objets étudiés : la minimisation des croisements qui a entraîné la création des tracés planaires et orthogonaux est étroitement liée aux représentations en VLSI (*Very Large Scale Integration*, qui est une technologie de circuit intégré) où trop de croisements peuvent notamment entraîner des problèmes thermiques [168]. Dans le cadre des réseaux sociaux, un ensemble de nœuds proches dans l’espace de plongement peut s’interpréter comme une communauté, ou une position sociale peut être mise en évidence en observant un ensemble d’acteurs dont les connexions au reste du réseau se comportent de façon similaire.

Du fait de la complexité des problèmes, qui sont bien souvent NP-difficiles, et de leur importance applicative, de très nombreuses approches ont été proposées (e.g. [16]). Néanmoins, dans l’étendue de la littérature, on peut dégager plusieurs approches parmi les plus couramment utilisées : les tracés orthogonaux [64], les tracés circulaires [107, 102], les tracés de type “force et ressort” qui ne reposent pas sur des contraintes *a priori* comme les précédentes mais sont le résultats d’une approche qui fournit des formes assez caractéristiques, et les tracés de type “arbre”. Dans la suite de cette section, nous nous concentrons sur les tracés de type force et ressort ainsi que sur les premières tentatives qui ont été proposées pour représenter les tracés non pas dans le plan, mais dans un espace 3D.

3.2.2 Tracés de type force et ressort

Les tracés appartenant à la famille des approches force et ressort reposent tous sur une analogie physique pour régir le placement des nœuds dans l’espace. Typiquement, ces approches utilisent une fonction modélisant une force attractive qui s’efforce de rapprocher les nœuds connectés et une force répulsive entre toutes les autres paires de nœuds. Cela consiste alors à trouver un tracé pour lequel la somme des forces sur chacun des nœuds est nulle. Formulé autrement, il s’agit de trouver une optimisation du tracé pour laquelle la somme des forces nettes soit minimale. Une approche de ce type comporte donc un modèle “énergétique” et une fonction d’optimisation qui lui est associée.

Les premières mentions de ce type de tracés remontent aux années 60, avec les travaux de Fisk et Isett qui proposent une méthode permettant de dessiner un circuit électronique en se basant sur un modèle représentant les liens entre les composants

par des élastiques. Mais il faut attendre les années 80 et les travaux pionniers de Eades [60] pour que ces méthodes commencent à susciter l'intérêt de la communauté. Dans le modèle qu'il propose (appelé *Spring Embedder*), les liens attirent les nœuds à leurs extrémités en suivant une loi logarithmique modélisant le comportement d'un ressort tandis que tous les nœuds se repoussent entre eux via une loi en carré inverse, simulant ainsi des particules chargées électriquement. A ce modèle de forces, Eades associe une fonction d'optimisation itérative qui s'applique à un tracé initial aléatoire des nœuds. A partir de cette position, les nœuds sont déplacés simultanément en fonction des forces qui s'exercent sur eux.

Depuis ce modèle, de nombreuses modifications ont été proposées dans la littérature pour améliorer les performances des calculs et la qualité du tracé final. Une des principales contributions provient certainement des travaux de Fruchterman et Reingold [78] qui ont enrichi le modèle initial en modifiant les forces régissant les placements afin d'accélérer les calculs. En sus, ils proposent différentes optimisations comme l'ajout d'un paramètre d'atténuation des forces qui limite les déplacements lorsque le nombre d'itérations croît et une distance seuil au delà de laquelle les forces répulsives n'agissent plus. Ces modifications entraînent une convergence plus rapide de l'algorithme vers un tracé pour lequel les forces s'équilibrent. Cependant, il n'y a pas d'assurance que le résultat soit un optimum global. D'autres variations de cet algorithme ont été proposées permettant, par exemple, d'éviter le chevauchement de nœuds, de fixer la position de certains nœuds ou encore de traiter des communautés identifiées au préalable de manière indépendante [197, 93].

Une autre contribution majeure provient des travaux de Kamada et Kawai [110] qui proposent d'utiliser un seul type de force entre tous les nœuds : la loi de Hooke qui décrit les forces exercées par un ressort entre ses deux extrémités. A chaque paire de nœuds, qu'ils soient liés ou non, est associée un ressort caractérisé par sa longueur au repos qui exerce une force proportionnelle à sa déformation : les nœuds sont attirés l'un vers l'autre si la distance qui les sépare est plus grande que la longueur du ressort au repos. Dans le cas contraire, ils se repoussent. La longueur au repos des ressorts est définie en fonction de la taille de la chaîne la plus courte entre les nœuds qui le composent. Contrairement aux approches citées ci-dessus, l'optimisation du modèle ne consiste pas à déplacer les nœuds simultanément mais à en déplacer un seul à la fois. Quelques années plus tard, Cohen [42] note que cette approche comporte de nombreuses similitudes avec le "*Multidimensional scaling*", ce qui permet d'employer des heuristiques d'optimisation assez efficaces basées sur la minimisation du stress [81].

Les tracés émergents de ces approches sont très populaires pour différentes raisons. Tout d'abord, ils sont souvent considérés comme capables de produire des tracés de bonne qualité pour des graphes de taille moyenne. En effet, comme l'a montré Noack [148], ces tracés produisent un partitionnement maximisant la modu-

larité. De plus, l'analogie avec des forces réelles rend ces algorithmes intuitivement appréhendables. Et, ils mettent généralement bien en évidence certaines structures intéressantes des graphes comme les symétries ou les amas de nœuds [63]. Pour ces raisons, auxquelles s'ajoute ici l'invariance de l'algorithme selon les conditions expérimentales (2D, 3D monoscopique et 3D stéréoscopique) nous avons sélectionné cette famille d'approches (et en particulier l'algorithme de [78]) pour nos expérimentations. En effet, comme le soulignent Herman et al. [96], les approches reposant sur une analogie de forces [60, 47] peuvent, d'une manière générale, être exprimées dans un espace 3D comme dans un espace 2D, sans modifications particulières.

3.2.3 Tracés 3D monoscopique

A notre connaissance les premiers travaux théoriques concernant les représentations 3D d'un graphe peuvent être attribués à Kolmogorov et Barzdin en 1967 [119]. Il faudra cependant attendre les années 80 et la formulation des premiers problèmes concrets pour que l'étude des tracés 3D attire l'attention de la communauté scientifique. Par exemple, en 1983, Rosenberg [168] étudie les avantages théoriques *a priori* de circuits imprimés en 3D sur plusieurs couches.

Dès le début des années 90, des systèmes permettant de tracer et de visualiser des graphes en 3D ont été proposés dans la littérature. L'un de ces précurseurs est le système des *Cone Trees* développé par Robertson en 1991 [166]. Il permet de visualiser une classe particulière des graphes : les arbres, pour en dévoiler les structures hiérarchiques. La racine de chaque hiérarchie est représentée au sommet d'un cone tandis que ses enfants sont distribués de façon uniforme à la base du cone. Il faut ensuite attendre le milieu des années 90 pour que les premières approches visant à visualiser des graphes génériques commencent à voir le jour (par exemple [154]). Contrairement aux *Cone Trees* qui emploient une métaphore visuelle spécifique à un espace 3D, la majorité des systèmes proposés repose sur une généralisation d'un tracé 2D à un espace 3D. En 1995 par exemple, Bruss et al. proposent le système *Gem-3D* permettant de visualiser des graphes en utilisant une variante 3D du *Spring-Embedder*.

Dans la suite des années 90, une grande partie des travaux de la communauté *Graph Drawing* s'est concentrée sur l'adaptation des algorithmes les plus populaires à un espace 3D comme les tracés orthogonaux, très utilisés de part les propriétés qu'ils partagent avec le VLSI. Ainsi, plusieurs adaptations du tracé orthogonal à un espace 3D ont été proposées [42, 14, 207, 24]. Les positions des nœuds sont contraintes aux croisements des lignes d'une grille tri-dimensionnelle et les liens sont représentés sur son maillage.

En 1996, Eades [64] reformule les travaux de Kolmogorov et Barzdin pour que

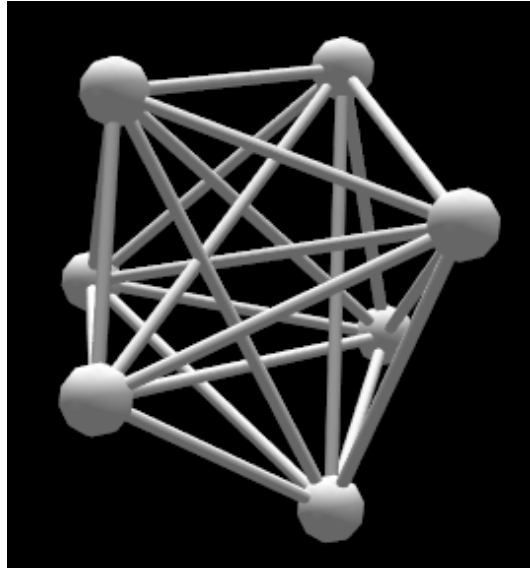


FIGURE 3.3 – Illustration d’un tracé du graphe K_7 . Le tracé du graphe a été obtenu via un algorithme de type force et ressort. Quel que soit le point de vue choisi, le nombre de croisements de liens perçu est toujours important.

les tracés résultants soient conformes aux critères esthétiques les plus répandus. Cette contribution illustre l’état d’esprit dans lequel les tracés 3D ont été abordés ; l’objectif était d’optimiser des critères esthétiques mis en évidence dans des tracés 2D. Le bien fondé de cette approche sort du cadre de cette thèse mais nous pouvons cependant noter que cela a eu pour conséquence l’émergence de ce que la communauté considère comme les deux problèmes principaux des tracés 3D [96] : (i) en 3D, des objets peuvent en cacher d’autres et (ii) il est difficile de choisir le meilleur angle de vue sur le tracé dans un espace 3D. Le tracé du graphe K_7 est paradigmatique de ces problèmes (voir la figure 3.3). Soulignons toutefois l’existence de deux autres approches ayant trait aux espaces 3D. Tout d’abord les tracés dans des espaces hyperboliques [141] qui permettent de visualiser des hiérarchies contraintes à une sphère. Dans cette approche, la racine de la hiérarchie est située au centre de la sphère, les branches de la hiérarchie sont disposées à l’intérieur de la sphère tandis que les enfants sont répartis sur sa surface. Une autre approche proposée par Fekete et al. [69] consiste à représenter les nœuds par des objets 2D (e.g. des carrés) placés de manière parallèle au plan (x, y) dans un espace 3D. Deux nœuds sont alors liés s’ils sont superposés (e.g. un carré en cache un autre) : il existe alors une droite parallèle à l’axe z passant par les objets les représentant.

3.3 Interactions en fouille de graphes

La restitution graphique d’un tracé de graphe ne constitue que la première étape de la boucle de raisonnement proposée par Keim et rappelée dans l’introduction

de ce chapitre. Le système employé doit pouvoir fournir différents outils à l'utilisateur pour lui permettre de diriger l'exploration des données et d'en extraire des connaissances pertinentes. En 1996, Schneiderman a défini ce processus interactif avec le mantra que nous avons déjà mentionné : “*Overview first, Filter and zoom, Details on demand*”. Ce dernier a été complété plus récemment par Keim pour devenir “*Analyze first, Show the Important, Zoom, filter and analyze further, Details on demand*” [99, 113].

Comme nous l'avons déjà précisé section 3.1, la première phase d'analyse permet de sélectionner les données sur lesquelles travailler ainsi que d'identifier le type de connaissances que l'utilisateur souhaite acquérir. Ensuite, la première étape de la visualisation consiste à restituer une vue d'ensemble du graphe en s'affranchissant des détails peu porteurs d'informations, comme par exemple des nœuds ayant un degré faible.

Quelque soit l'approche de tracé choisie, un problème inhérent à la visualisation d'informations émerge rapidement quand le nombre d'acteurs et le nombre de liens entre ces acteurs croît : l'occlusion et les croisements entre les liens rendent l'affichage graphique trop chargé et il devient compliqué d'extraire des informations du processus de visualisation. Plusieurs procédés sont rapportés dans la littérature pour contourner cette difficulté. Par exemple, on peut procéder à l'agrégation des nœuds d'un sous-graphe en regroupant ceux qui sont les plus proches dans un méta-nœud [5]. Une méthode plus récente consiste à regrouper non pas les nœuds, mais les liens via un procédé nommé “*Edge Bundling*” [190].

Ces méthodes capturent efficacement la structuration globale du graphe mais entraînent inévitablement une perte d'information à un niveau de granularité plus petit. Ce problème du “Focus & Context”, bien connu de la communauté de la visualisation de l'information, nécessite la mise en œuvre d'approches permettant l'observation de propriétés structurelles à différentes échelles.

Les outils interactifs deviennent alors primordiaux pour surmonter cette difficulté et explorer le graphe à différents niveaux de granularité. Pour y parvenir, la plupart des systèmes de fouille de graphes proposent des mécanismes de zoom permettant d'agrandir la restitution de régions spécifiques du graphe. Cependant, une fois le zoom effectué sur une partie du graphe, cette dernière occupe le plus souvent une partie importante de l'espace d'affichage et il est donc difficile de la mettre en perspective avec le reste du graphe. Certaines alternatives ont donc été développées ; par exemple l'émulation d'une lentille à grand angle (*Fisheye*) grossissant les objets situés directement sous le pointeur de la souris tandis que les objets plus éloignés apparaissent de plus en plus petits [174]. Une autre approche consiste à produire plusieurs vues sur les données [15] en ayant par exemple une fenêtre représentant l'ensemble du graphe et une autre représentant la partie sur laquelle l'utilisateur a zoomé.

Quel que soit le niveau de granularité auquel le graphe est visualisé, certains nœuds et liens peu porteurs d'informations peuvent entraver la fouille (par exemple des nœuds isolés ou ne correspondant pas à certaines propriétés). Il est donc important de proposer à l'utilisateur des mécanismes de filtrage permettant de n'afficher que les nœuds et liens susceptibles d'être intéressants. L'outil *Visone* [29] permet par exemple de filtrer manuellement la restitution du graphe : c'est l'utilisateur qui sélectionne les nœuds à cacher ou à modifier via un menu. Le filtrage peut aussi être semi-supervisé comme dans [158], où l'utilisateur peut effectuer un seuillage par rapport à différentes métriques afin de restreindre le nombre de nœuds et de liens affichés simultanément.

A tout moment, l'utilisateur doit également pouvoir consulter les informations sémantiques associées aux nœuds et aux liens. Dans la plupart des systèmes, ces informations sont cachées de façon à ne pas surcharger l'affichage du graphe. Elles sont alors accessibles en sélectionnant le nœud ou le lien d'intérêt.

En sus de ces interactions, notons que certaines mesures spécifiques doivent être prises dans le cas d'une visualisation 3D. Comme nous l'avons souligné dans la section précédente, la plupart des restitutions 3D entraînent de l'occlusion dans le tracé. L'ensemble du graphe n'est donc pas nécessairement visible depuis la vue initiale et l'utilisateur doit donc pouvoir explorer l'espace pour le visualiser sous différents angles de vue.

TempoSpring : un environnement logiciel expérimental pour la fouille de graphes en 3D

The idea behind digital computers may be explained by saying that these machines are intended to carry out any operations which could be done by a human computer.
—Alan Turing

Sommaire

Introduction	56
4.1 Architecture générale	56
4.1.1 Algorithme de tracé	58
4.1.2 Optimisation des calculs	61
4.1.3 Implémentation de la stéréoscopie	63
4.2 Mécanismes d'interaction	63
4.3 Graphes Dynamiques	67
4.3.1 Présentation des principales méthodes d'intégration de la dynamique	68
4.3.2 La dynamique dans <i>TempoSpring</i>	72

Introduction

L'évaluation de la stéréoscopie en fouille de graphes nécessite un environnement expérimental capable d'afficher des graphes de tailles moyennes (quelques centaines de nœuds) sous différentes conditions de restitutions. Notons que nous n'avons pas abordé ici les restitutions de très grands graphes qui nécessitent la combinaison d'opérateurs dont la transposition du 2D au 3D posent de nombreuses questions à notre connaissance largement ouvertes. De nombreux outils permettent de visualiser des graphes (voir ce catalogue [161]) mais aucun, à notre connaissance, ne propose à la fois des restitutions 2D, 3D monoscopique et 3D stéréoscopique. De plus, de nombreuses caractéristiques techniques diffèrent entre ces outils (format d'entrée, paramétrage des algorithmes de tracé, choix graphiques etc ...) : en utiliser un par méthode de restitution aurait donc entraîné des biais inévitables dans la comparaison. Pour ces raisons, nous avons développé un nouvel environnement expérimental : *TempoSpring*, que nous présentons dans ce chapitre.

Dans la première section, nous décrivons l'architecture générale de *TempoSpring* en énumérant les principaux composants du système. Nous détaillons ensuite l'algorithme de force et ressort que nous avons choisi pour le tracé et les modifications que nous lui avons apportées. Après avoir rappelé les principes de la restitution stéréoscopique implémentée dans l'outil, nous présentons également les mécanismes d'optimisation mis en œuvre : une optimisation de type *Octree* et le calcul parallèle sur carte graphique. Dans la deuxième section, nous abordons les mécanismes proposés par *TempoSpring* pour le processus de fouille visuelle interactif. Ces mécanismes permettent d'explorer l'espace de visualisation, d'accéder au contenu sémantique des nœuds, d'identifier les nœuds et les liens importants, de filtrer l'affichage et d'agréger manuellement des groupes de nœuds. Nous terminons ce chapitre en évoquant la visualisation dynamique de graphes qui, sans avoir été validée expérimentalement, est prise en charge par notre outil. A travers un bref état de l'art, nous dressons les grandes lignes de la visualisation dynamique de graphes puis nous exposons les mécanismes retenus pour son implémentation dans *TempoSpring*.

4.1 Architecture générale

L'environnement *TempoSpring* s'appuie sur une architecture principalement développée en C++ et *OpenGL*. Un diagramme de classes simplifié représentant les différentes briques logicielles de *TempoSpring* est proposé sur la figure 4.1. Le système *TempoSpring* s'articule autour de cinq paquets principaux :

- **OpenUtility** est une bibliothèque graphique développée au sein de l'équipe de recherche *COnnaissance et Décision* [91]. Ce paquetage comporte plu-

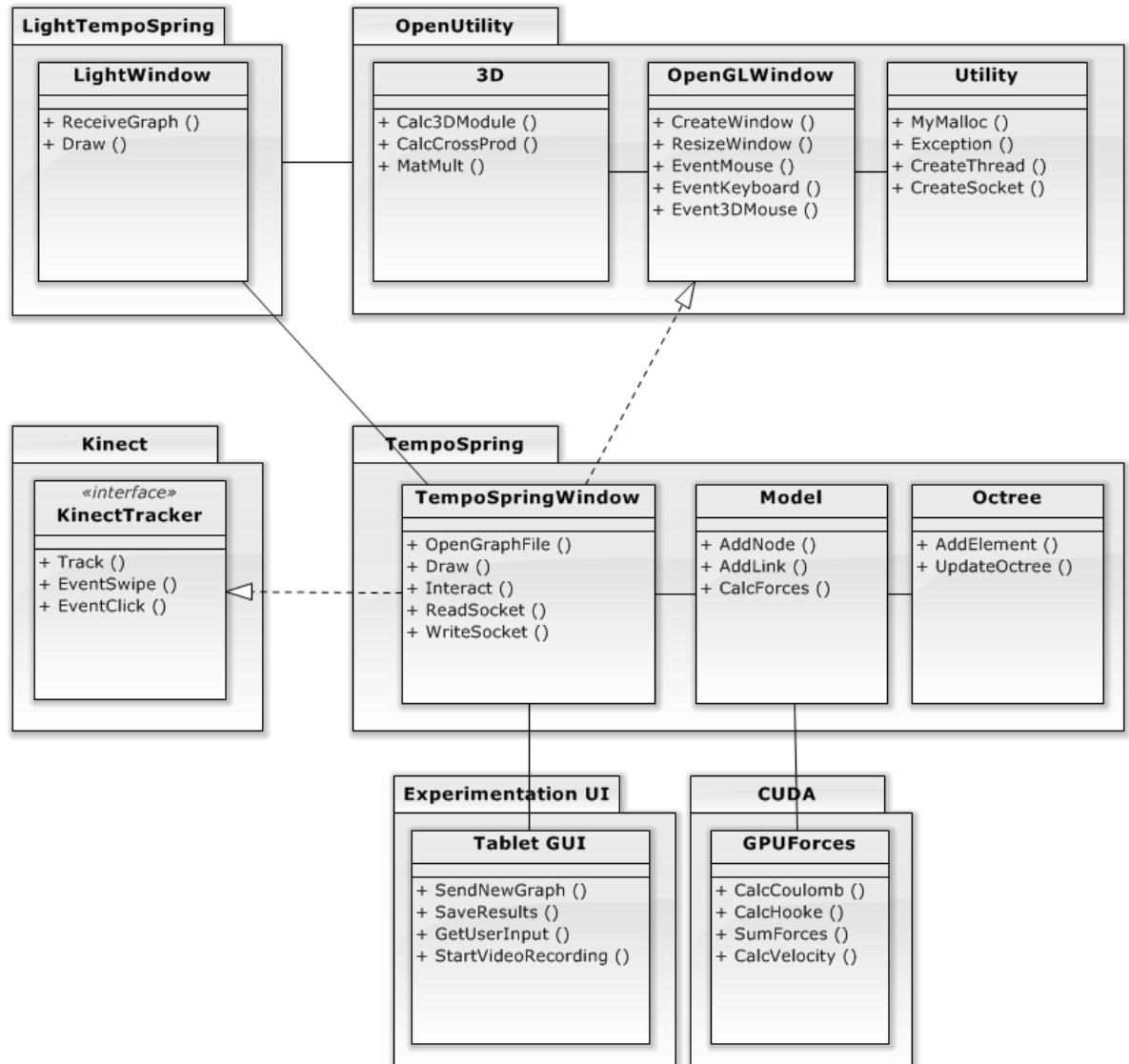


FIGURE 4.1 – Diagramme UML simplifié de *TempoSpring*. Seules les classes principales sont représentées sur ce diagramme. Pour ces classes, des méthodes illustrant leur fonctionnement sont également incluses. Ce système s’articule autour de deux paquets centraux : **TempoSpring** et **OpenUtility**. Le premier gère la visualisation des graphes et les interactions en s’appuyant sur les bibliothèques fournies par le second. La classe principale **TempoSpringWindow** gère également les communications réseaux pour pouvoir communiquer avec la version allégée de *TempoSpring* (**LightTempoSpring**) et avec une interface tablette (**Tablet GUI**) permettant de déclencher certaines fonctionnalités (voir les chapitres 5 et 7).

sieurs classes utilitaires permettant de faciliter la création d'applications de visualisation 3D. Sur le diagramme UML, les trois classes principales sont représentées. La classe **3D** propose de nombreuses fonctions facilitant les calculs matriciels et vectoriels inhérents aux applications graphiques 3D. La classe **OpenGLWindow** facilite la création et la gestion de fenêtres *OpenGL*. La classe **Utility** facilite la gestion des mécanismes de bas niveau comme les allocations mémoires, les exceptions ou la création de canaux de communications réseaux et le *multithreading*.

- **TempoSpring** est le cœur du système et s'appuie sur le paquetage **OpenUtility**. Ce paquetage permet d'afficher des graphes à partir d'un format d'entrée très simple (une illustration est présentée sur la figure 4.2) et d'interagir avec ceux-ci. La classe principale, **TempoSpringWindow** permet l'ouverture des graphes, leur affichage, gère les interactions et permet de communiquer en réseau avec d'autres composants du système. La classe **Model** maintient la représentation interne du graphe et gère les calculs relatifs au tracé du graphe. La classe **Octree** permet d'altérer la représentation interne via l'utilisation d'une structure de type *octree* afin d'optimiser le calcul des forces. Les calculs peuvent également être optimisés en faisant appel à des mécanismes d'accélération matériel via le paquetage **CUDA** et la classe **GPUForces**. Ces deux points sont détaillés dans la section 4.1.2.
- **ExperimentationUI** est un paquetage *JAVA* qui permet d'utiliser une interface graphique simple pour contrôler certains aspects de *TempoSpring* comme l'ouverture d'un graphe spécifique. Cette interface peut être lancée depuis une machine située sur le même réseau que la machine exécutant *TempoSpring*. Le fonctionnement de cette interface est décrit en détail dans le chapitre 5.
- **LightTempoSpring** est un paquetage *JAVA* et *OpenGL ES* qui agit comme une version allégée de *TempoSpring* en s'appuyant sur une version simplifiée de la bibliothèque **OpenUtility**. La classe principale **LightWindow** permet d'afficher des graphes qui lui sont envoyés par **TempoSpringWindow** et dont le tracé a déjà été calculé. Cette partie du système est détaillée dans le chapitre 7.
- **Kinect** est un paquetage permettant de faire l'interface entre les pilotes du capteur *Kinect* et *TempoSpring* pour convertir des gestes en interactions. Son fonctionnement est décrit en détail dans le chapitre 7.

4.1.1 Algorithme de tracé

Comme nous l'avons précisé dans le chapitre 3, notre choix d'algorithme de tracé s'est porté sur le modèle force et ressort de Fruchterman et Reingold [78] basé sur la loi de Hooke (approximation macroscopique de la force attractive ap-

```

1 Nodes
2 {
3 1:1223:1253:1:(255,255,255):"Robert G #6158 (1223-1253)";
4 2:1238:1278:1:(255,255,255):"Pelissier B #3331 (1238-1278)";
5 ...
6 }
7 Edges
8 {
9 2:3:1:yellow;
10 2:1:1:blue;
11 ...
12 }

```

FIGURE 4.2 – Illustration du format d’entrée de *TempoSpring*. Un nœud (resp. lien) est représenté par une ligne dans le bloc *Nodes* (resp. *Edges*). Les données qui leur sont associées sont séparées par le caractère “:”. Pour les nœuds, ces données correspondent à : “*identifiant : temps-début : temps-fin : charge : couleur : Données sémantiques* ;” et pour les liens : “*identifiant1 : identifiant2 : longueur : couleur* ;”. La charge des nœuds et la longueur des liens sont des paramètres relatifs à l’algorithme de tracé. Le contenu sémantique, qui peut être sous forme d’images ou de texte, est affiché lorsque l’utilisateur sélectionne un nœud.

pliquée par un ressort entre deux objets) entre les nœuds liés et la loi de Coulomb (modélisation des forces électromagnétiques agissant entre particules chargées électromagnétiquement) entre toutes les paires de nœuds.

L’attraction de Hooke (équation 4.1) est appliquée à un nœud de coordonnées p_1 lié à un nœud de coordonnées p_2 avec k la constante de Hooke (ou constante de raideur, qui détermine la rigidité du ressort) et R la longueur à vide du lien. Entre chaque paire de nœuds de coordonnées p_1 et p_2 , la répulsion de Coulomb (équation 4.2) est appliquée avec ϵ_0 la constante électrique et q_1 et q_2 , les charges électriques de ces particules. Toutes les variables sont extraites des données hormis les paramètres ϵ_0 et k qui peuvent être modifiés en temps réel afin de faire varier l’importance des forces. Cette variation entraîne un éloignement ou un rapprochement des nœuds entre eux.

Tout comme dans l’approche de Eades, la fonction d’optimisation visant à minimiser l’énergie totale dans ce système de particules et de ressorts dépend du déplacement des nœuds à partir d’un tracé initial aléatoire. Pour chaque nœud, les forces s’exerçant sur lui déterminent sa vitesse. Une fois que toutes les vitesses de tous les nœuds sont connues, les nœuds sont déplacés. Lors de ce déplacement, nous calculons le barycentre du graphe pour pouvoir le maintenir au centre de l’écran évitant ainsi que le graphe ne se déplace en dehors du point de vue de l’utilisateur. Par défaut, un pré-traitement est appliqué sur le graphe afin d’en extraire sa composante connexe de plus grande taille qui sera affichée (les autres composantes connexes peuvent être sélectionnées de manière interactive).

Un résumé du fonctionnement de l’algorithme est présenté sous forme de *pseudocode* (algo. 1). Par défaut, la restitution du tracé est constamment mise à jour

(après chaque itération de la partie *main loop* de l’algorithme 1), mais il est possible de spécifier des positions fixes pour chacun des nœuds *a priori*. Dans ce cas, l’algorithme de tracé n’est pas utilisé.

Les nœuds du graphe sont alors représentés par des sphères et les liens sont représentés par des cylindres. Notre choix s’est porté sur des cylindres et non sur de simples lignes pour deux raisons : (i) la nature tridimensionnelle d’un cylindre facilite sa perception et (ii) leur aspect graphique est moins sensible à l’effet escalier (*aliasing* ou crénelage) rendant les contours plus lisses et améliorant ainsi la visibilité de la restitution.

$$F = -k \left(\frac{(|L| - R)L}{|L|} \right) \text{ avec } L = p_2 - p_1 \quad (4.1)$$

$$F = \frac{1}{4\pi\epsilon_0} \frac{q_1 q_2 (p_1 - p_2)}{|p_1 - p_2|^3} \quad (4.2)$$

Data : ListOfNodes

Result : Converging graph layout

initialization;

for each node n do

$n.\text{position} = \text{randomize}();$

end

$damping \leftarrow 0.2;$

main loop;

for each node n do

$n.\text{netForce} \leftarrow 0;$

for each other node m do

$n.\text{netForce} \leftarrow n.\text{netForce} + \text{coulomb}(n, m);$

if linked(n, m) then

$n.\text{netForce} += \text{hooke}(n, m);$

end

end

$n.\text{velocity} \leftarrow n.\text{velocity} + (n.\text{netForce} \times damping);$

end

for each node n do

$n.\text{Position} \leftarrow n.\text{Position} + n.\text{velocity};$

end

Algorithme 1 : Pseudocode de l’algorithme de tracé utilisé par l’outil *TempoS-pring*. Les fonctions “coulomb” et “hooke” correspondent aux équations 4.1 et 4.2. La variable “damping” permet de limiter la vitesse des nœuds de façon à ne pas apporter de changements trop abrupts du tracé (qui est mis à jour en temps réel à l’écran).

4.1.2 Optimisation des calculs

La méthode de tracé que nous utilisons a une complexité de $O(n^2)$. Cette complexité la rend inutilisable dès lors que les graphes sont de trop grande taille. Nous avons donc mis en place deux mécanismes mutuellement exclusifs permettant d’optimiser les calculs de l’algorithme de tracé : (i) une optimisation logicielle de type *Octree* et (ii) une optimisation matérielle via utilisation de l’architecture parallèle des cartes graphiques.

4.1.2.1 Octree

Cette méthode est inspirée du problème des simulations *N-corps* en physique expérimentale visant à résoudre les équations du mouvement de Newton pour de nombreux corps interagissant gravitationnellement entre eux. Barnes et Hut [12] ont proposé cette méthode en 1986 pour passer d’une complexité $O(n^2)$ à $O(n \log n)$. Brièvement, il s’agit de découper l’espace de simulation via une structure hiérarchique tridimensionnelle appelée *Octree*. La racine de cette hiérarchie est le cube représentant le volume de simulation. Elle a huit enfants qui correspondent chacun à $1/8e$ du volume. Par récursivité, ces enfants sont également divisés en huit autres enfants et ainsi de suite. La finesse du calcul de force dépend alors de la distance de deux éléments dans la hiérarchie. A l’intérieur d’une cellule sans enfant, les calculs se font de manière “pair à pair”, tandis qu’entre les cellules, on applique une force issue de leur centre de gravité. Nous avons implémenté cette approche dans *TempoSpring* (une illustration avec le détail de l’octree ajouté à la restitution est présentée sur la figure 4.3).

4.1.2.2 Calculs GPU

Après chaque itération de l’algorithme de tracé, il est nécessaire de mettre à jour les positions des nœuds dans l’*Octree* en fonction de leur position dans l’espace d’affichage. Rapidement (pour des graphes de plus de 400 nœuds), ce coût de maintien d’un *Octree* devient trop élevé et l’affichage en temps réel perd en fluidité. Pour cette raison, nous avons implémenté une version de l’algorithme de tracé utilisant le langage de programmation GPU *Cuda* [77, 10]. Ce langage exploite le grand nombre d’unités arithmétiques et logiques présentes dans les cartes graphiques modernes pour paralléliser les calculs qui s’en trouvent grandement accélérés. Pour notre implémentation de l’algorithme de tracé, c’est une partie modifiée du *main loop* qui est parallélisée. Brièvement, chacune des itérations peut être exécutée indépendamment puisque toutes les positions sont mises à jours en même temps une fois que la vélocité a été calculée. Dans le langage *Cuda*, ces itérations sont calculées sur des *threads* (unités de calcul indépendantes). Sur la carte gra-

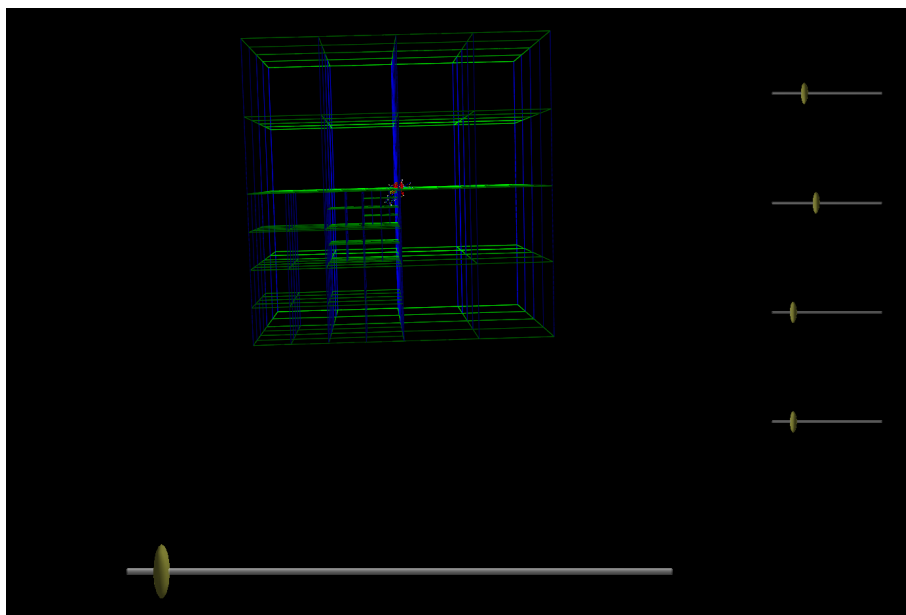


FIGURE 4.3 – Illustration de l’optimisation de type *Octree* dans *TempoSpring*. L’espace est divisé en une hiérarchie de cubes imbriqués. A l’intérieur d’un cube, les forces sont calculées de façon paire à paire puis agrégées (moyenne) et propagées vers le reste de la hiérarchie. Les calculs entre cubes ne prennent alors en compte que le centre de gravité des autres cellules. Sur cette image, le tracé du graphe vient juste d’être initialisé et une représentation de l’octree est proposée. La hiérarchie est plus profonde au niveau de l’espace englobant le tracé car nous développons un cube (subdivision d’un cube en huit sous-cubes) uniquement s’il contient plus de 50 nœuds.

phique utilisée pendant nos expérimentations, ces *threads* sont au nombre de 512. Il peut donc y avoir jusqu'à 512 itérations calculées en parallèle ce qui nous permet d'afficher des graphes ayant jusqu'à 3000 nœuds, tout en gardant une restitution fluide. Cependant, cette optimisation a un inconvénient majeur : les accès mémoire et les transferts d'information entre le CPU et le GPU (initialisation des positions des nœuds) sont très coûteux et rendent l'utilisation d'une structure de type *Octree*, qui demande beaucoup d'accès mémoires, très peu efficace. Il convient donc de choisir entre les deux approches en fonction du type de graphe visualisé.

4.1.3 Implémentation de la stéréoscopie

L'implémentation de la stéréoscopie dans *TempoSpring* suit les préconisations de [83]. Succintement, la production d'images virtuelles 3D consiste à afficher sur l'écran tout ou partie d'une région d'un espace dans lequel le monde virtuel est représenté. La forme exacte de cette région est fonction du point de vue (ou caméra) utilisé. Cette notion est approfondie dans le chapitre 6. Le plus souvent cette région a la forme d'un tronc de pyramide rectangulaire (*frustum*). Deux plans perpendiculaires à la direction du point de vue coupent ce tronc : le "plan proche" (*near plane*) et le "plan éloigné" (*far plane*). Seuls les objets inclus dans ce tronc sont restitués. Une illustration de projection monoscopique est disponible sur la figure 4.4.

Dans une restitution monoscopique un seul de ces troncs est utilisé et l'origine de la pyramide dont il est issu correspond à la position du point de vue. *A contrario*, en stéréoscopie on utilise deux troncs dont les pyramides ont pour origine deux points de vue différents situés à la droite et à la gauche d'un point de vue identique en monoscopie. Les directions des points de vue sont parallèles mais les troncs correspondant ne sont pas symétriques : ils sont construits de façon à converger (voir figure 4.5). La scène 3D est alors calculée pour ces deux points de vue et il devient possible de produire l'illusion de la stéréoscopie en ne rendant visible qu'un seul de ces points de vue par œil (point de vue droit avec œil droit et inversement). Une technologie récente offre cette possibilité en affichant les images droites et gauches entrelacées à une fréquence de $120Hz$. En utilisant des lunettes à obturation synchronisées avec le périphérique d'affichage, il est alors possible d'obtenir un des verres des lunettes toutes les deux images pour rendre la restitution stéréoscopique.

4.2 Mécanismes d'interaction

TempoSpring a été développé dans une optique expérimentale, mais intègre plusieurs mécanismes permettant de faciliter le processus de fouille visuelle. Cette section liste ces différents mécanismes conformes au mantra de la fouille visuelle :

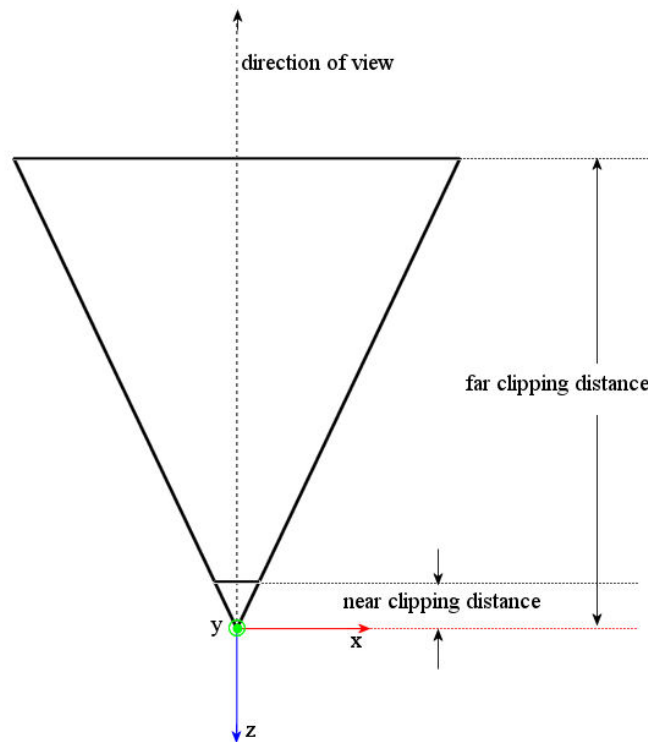


FIGURE 4.4 – Illustration de la projection monoscopique. Vue orthographique du dessus d’une projection monoscopique. Sur cette image, les *near clipping distance* et *far clipping distance* correspondent respectivement aux distances entre le point de vue et le plan proche et entre le point de vue et le plan éloigné. Les objets inclus dans le tronc formé à partir de ces deux plans et la pyramide sont restitués à l’écran.

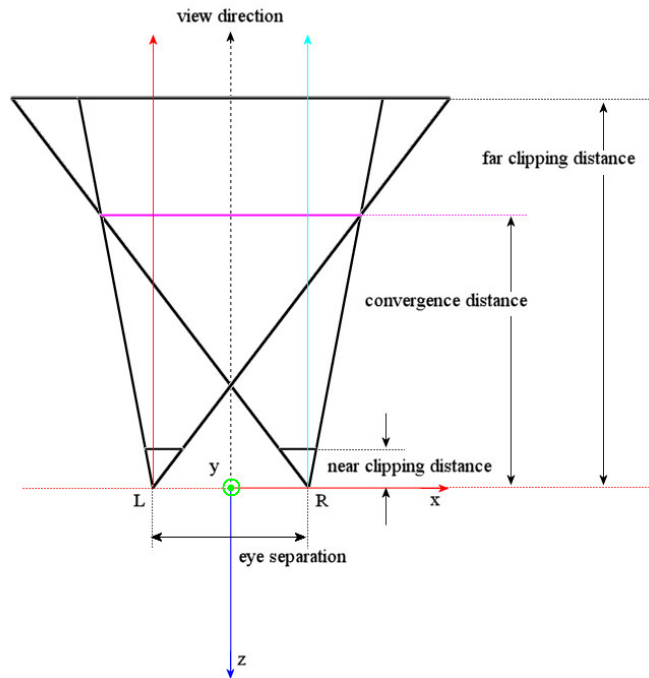


FIGURE 4.5 – Illustration de la projection stéréoscopique. Deux troncs non symétriques sont utilisés pour la projection stéréoscopique. Le plan sur lequel ils convergent structure la perception stéréoscopique. Les objets plus proches semblent apparaître “devant l’écran” tandis que les objets plus éloignés semblent apparaître “derrière l’écran”.

‘Analyze first, Show the Important, Zoom, filter and analyze further, Details on demand’.

Exploration de l’espace 3D Pour “contourner” les occlusions de certaines parties du tracé, il est nécessaire d’intégrer certains mécanismes permettant aux utilisateurs d’explorer l’espace d’affichage dans son ensemble. L’algorithme de tracé que nous utilisons assure que le graphe est centré à l’écran et, par conséquent, un utilisateur peut l’explorer via des rotations de type “arcball” [183] en déplaçant la souris. La molette de la souris est associée aux zooms avant et arrière qui permettent d’obtenir une vue d’ensemble ou une vue plus fine du tracé. Lors de l’affichage initial du graphe, le zoom est réglé automatiquement pour que l’ensemble du tracé soit visible à l’écran.

Accès au contenu sémantique En amont de la visualisation, différentes couleurs peuvent être associées aux nœuds et aux liens pour les différencier au besoin. Il est possible d’accéder au contenu sémantique des nœuds en cliquant dessus. Des informations qui peuvent être sous forme d’images ou de textes sont alors affichées dans un cadre dans la fenêtre de visualisation.

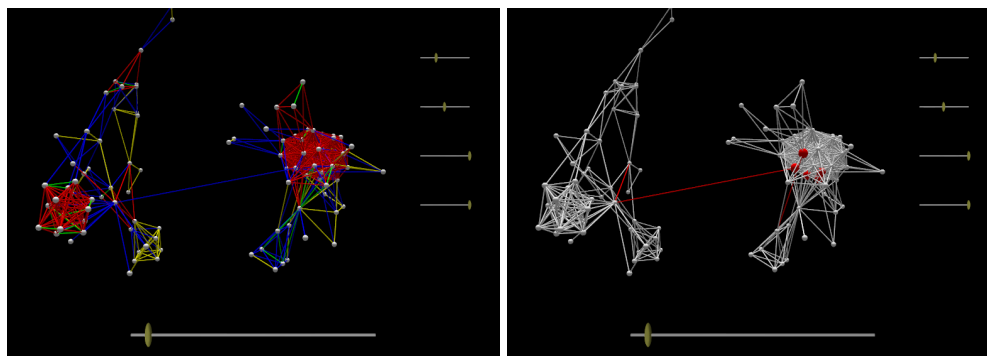


FIGURE 4.6 – Illustration du surlignage. Sur la première capture, nous pouvons voir l’intégralité du graphe avant filtrage. Sur la deuxième, les nœuds et les liens sujets aux forces les plus importantes sont mis en évidence : ils sont affichés en rouge tandis que le reste du graphe devient gris. Deux des curseurs situés à la droite de la fenêtre permettent de modifier le seuil de filtrage.

Identification des nœuds et liens importants Deux mécanismes d’aide à l’analyse ont été développés pour faciliter l’identification visuelle des nœuds et liens importants. C’est-à-dire des nœuds fortement connectés ou des liens faisant la jonction entre des groupes différents. Pour les nœuds, il s’agit de déterminer la somme absolue des forces exercées sur chacun des nœuds. Pour les liens, nous calculons le ratio de “tension”, qui est le rapport entre sa longueur dans la restitution et sa longueur à vide. Par défaut, seuls sont surlignés les nœuds (resp. liens) ayant une somme absolue des forces (resp. tension) supérieure à la valeur de la somme absolue des forces (resp. tension) du 95e centile de l’ensemble des nœuds (resp. liens). Un curseur permet de modifier ce seuil. Ce mécanisme est illustré par la figure 4.6. La somme absolue des forces permet d’identifier des individus centraux, qui sont fortement connectés au reste du réseau. La tension permet d’identifier des individus relais reliant des communautés peu connectées. Dans ce cas de figure, les forces répulsives entre leurs nœuds entraînent une forte tension sur les liens de ces individus relais.

Filtrage sur le degré des nœuds Ce filtre permet de cacher les nœuds et les liens qui lui sont attachés en fonction de son degré. Ce mécanisme illustré sur la figure 4.7 permet de n’afficher que le “squelette” du graphe afin d’aider à l’identification visuelle des hubs.

Agrégation manuelle des nœuds Cette fonctionnalité permet aux utilisateur d’agréger manuellement un ensemble de nœuds pour former un méta-nœud. Les méta-nœuds peuvent ensuite être agrégés entre eux et avec d’autres nœuds de manière hiérarchique. Ce mécanisme permet de simplifier la restitution en représentant les structures en communautés comme telles. La taille des méta-nœuds est une fonction du nombre d’éléments qu’ils contiennent et leur position correspond au barycentre

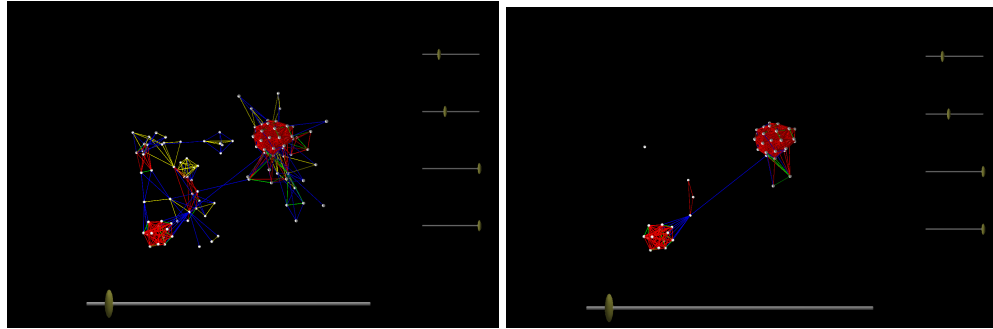


FIGURE 4.7 – Illustration du filtrage sur le degré des nœuds. Sur la première capture, nous pouvons voir l'intégralité du graphe avant filtrage. Sur la deuxième, les nœuds ayant un degré inférieur à 7 ont été cachés.

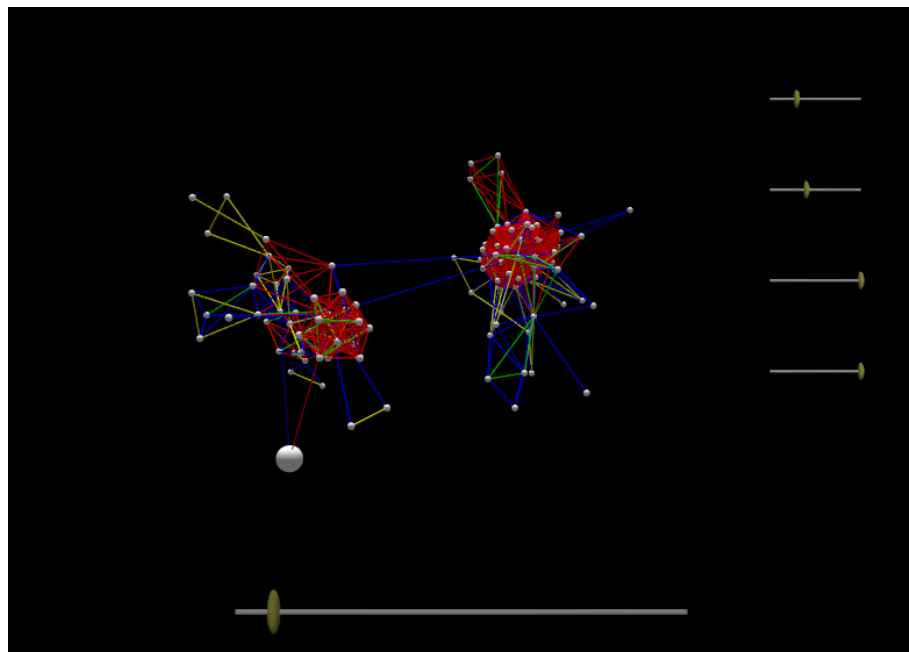


FIGURE 4.8 – Illustration de l'agrégation manuelle. Le nœud en bas à gauche de la capture représente un méta-nœud agrégé par un utilisateur.

des positions des nœuds contenus. Une illustration est présentée sur la figure [4.8](#)

4.3 Prise en compte des graphes dynamiques

TempoSpring a été conçu dès le début pour permettre le traitement de graphes dynamiques. Cette fonctionnalité a été testée empiriquement sur les données du projet ANR GRAPHCOMP présenté brièvement au chapitre 8 mais n'a pas encore été validée avec un protocole expérimental rigoureux.

4.3.1 Présentation des principales méthodes d'intégration de la dynamique

Dans cette section nous présentons trois approches proposées dans la littérature pour capturer la dimension temporelle d'un graphe dynamique. C'est à dire un graphe dont les nœuds et liens apparaissent et disparaissent avec le temps.

A notre connaissance, trois familles de métaphores visuelles se démarquent pour restituer graphiquement l'évolution :

- les résumés statistiques de graphes dynamiques : un ou plusieurs graphiques représentant l'évolution de certaines mesures au cours du temps sont affichés
- les séquences de graphes : des instantanés de graphes à des instants t différents sont affichés les uns à la suite des autres
- les animations de graphes qui sont des séquences de graphes animées

4.3.1.1 Modèle de temps orienté données

En préambule de la présentation des différentes métaphores visuelles, il est nécessaire de préciser certains points soulignés par Bender-DeMoll & McFarland dans [20]. En particulier, la fréquence d'échantillonnage des données relationnelles à visualiser impacte fortement la restitution finale puisqu'elle régule la façon dont le temps peut être représenté dans le système final.

Choisir un pas de temps entre les différents états successifs d'un graphe est certainement l'étape la plus délicate dans la confection d'un système de visualisation de graphes dynamiques. Imaginons par exemple un système de surveillance de courriers électroniques au sein d'une entreprise. Nous pouvons supposer que le trafic est moins important en dehors des heures de travail. Il serait donc pertinent de choisir un pas de temps de l'ordre de l'heure afin de pouvoir identifier ces périodes. En revanche, si le but du système est d'analyser des changements structurels (par exemple après une restructuration de l'entreprise), un pas de temps plus important serait requis pour pouvoir visualiser l'avant et l'après.

Deux stratégies permettent de capturer la dynamique [139, 192] : l'échantillonnage "continu" et l'échantillonnage "discret". Pour le cas "continu", on enregistre à partir de son état initial tous les événements qui interviennent [20]. Dans ce contexte, un événement peut être l'ajout ou la suppression d'un nœud ou d'un lien ou une variation d'un de leurs attributs. Pour le cas "discret", il s'agit d'enregistrer un instantané du graphe à chaque pas de temps. Notons qu'il est possible de transformer les données continues en représentation discrète : $G_{t+1} = G_t \vee Events_{t \rightarrow t+1}$, tandis que la réciproque ($Events_{t \rightarrow t+1} = G_t \setminus G_{t+1}$) n'est vraie que dans une certaine mesure (si la fréquence d'échantillonnage est plus importante que la fréquence

des évènements).

Dans le cas “discret”, la visualisation peut se ramener théoriquement à une séquence de problèmes de tracés mais la prise en compte de l’effort cognitif de retraitement nécessite des approches adaptées. Pour le cas “continu”, différentes stratégies ont été proposées [56], dont notamment :

- “*Sliding windows*” : le graphe est construit par l’agrégation de tous les évènements survenant durant une fenêtre temporelle spécifique. L’évolution est représentée en incrémentant l’instant auquel commence la fenêtre et l’instant auquel elle se termine.
- “*Expanding windows*” : le graphe est construit par l’agrégation de tous les évènements à partir d’un instant de départ, jusqu’à un curseur temporel représentant l’instant de fin. L’évolution est alors représentée en incrémentant le curseur temporel.

La première approche facilite la comparaison de deux graphes correspondant à deux périodes différentes et la deuxième est plus adaptée à la représentation de la série d’évènements qui ont conduit à l’état du graphe à l’instant correspondant à la position du curseur temporel.

4.3.1.2 Résumés statistiques de graphes dynamiques

Les premiers travaux ont considéré un graphe dynamique comme une succession de périodes pour lesquelles différentes métriques statistiques (*Q-Analysis*) sont calculées [56]. Ces métriques sont ensuite comparées les unes aux autres afin d’en extraire les caractéristiques de la dynamique.

Plus récemment, Falkowski et al. [67] se sont également intéressés à des métriques statistiques permettant de résumer l’évolution d’un graphe. Ils considèrent plus particulièrement la dynamique de différents sous-groupes dans un réseau évolutif. Ils procèdent tout d’abord à l’identification de ces sous-groupes via un processus de classification non supervisée. Ensuite ils calculent différents indicateurs comme la stabilité, la densité et la cohésion propres à chaque sous-groupe. La courbe de ces indicateurs en fonction du temps est ensuite tracée afin de pouvoir analyser leur évolution (figure 4.9).

Le formalisme employé permet une reproduction aisée. Nous pouvons toutefois identifier deux inconvénients majeurs de ces approches. Premièrement, elles ne capturent que les caractéristiques de granularité macroscopique (au niveau des communautés) et n’apportent pas d’indication sur les changements plus fins (au niveau des nœuds). Deuxièmement, elles ciblent des experts des réseaux sociaux capables de comprendre et d’interpréter les statistiques utilisées alors que cette probléma-

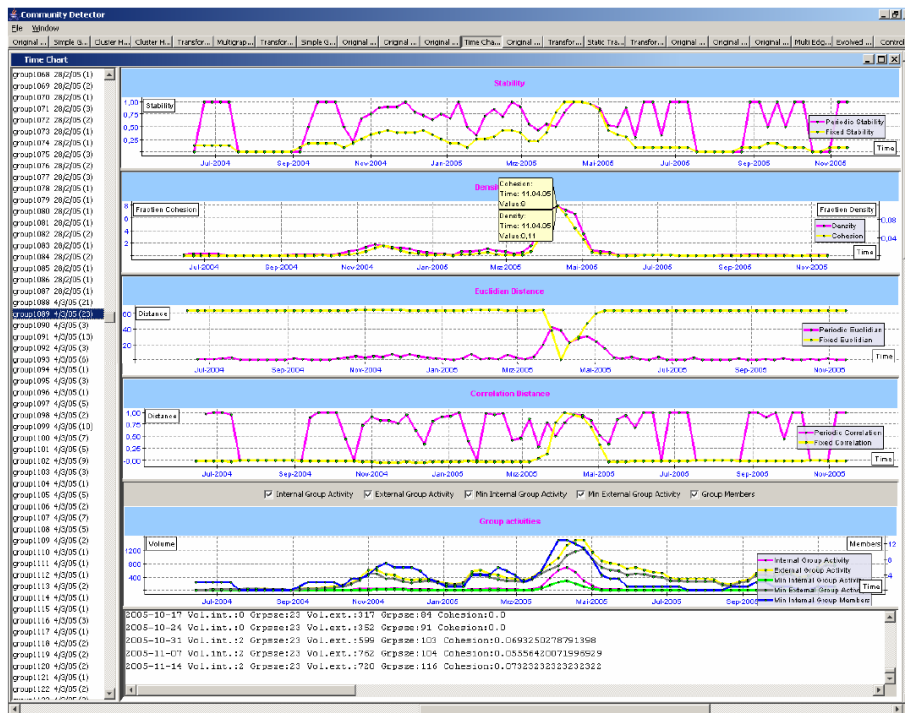


FIGURE 4.9 – Illustration des résumés statistiques de graphes dynamiques. Les courbes de différents indicateurs comme la cohésion, la stabilité ou la densité sont tracées en fonction du temps. Image provenant de [67].

tique sort de ce champ spécialisé (visualisation de données très spécifiques comme des graphes biologiques par exemple). Les innovations récentes en la matière vont d'ailleurs dans ce sens : Rosvall & Bergstrom [171] ont récemment présentés un système permettant de représenter les évolutions de réseaux sociaux sous forme de diagrammes alluviaux (figure 4.10). Ces diagrammes sont formés de plusieurs blocs représentant différentes communautés. Ils s'étendent ou rétrécissent en fonction des évolutions du réseau sous-jacent.

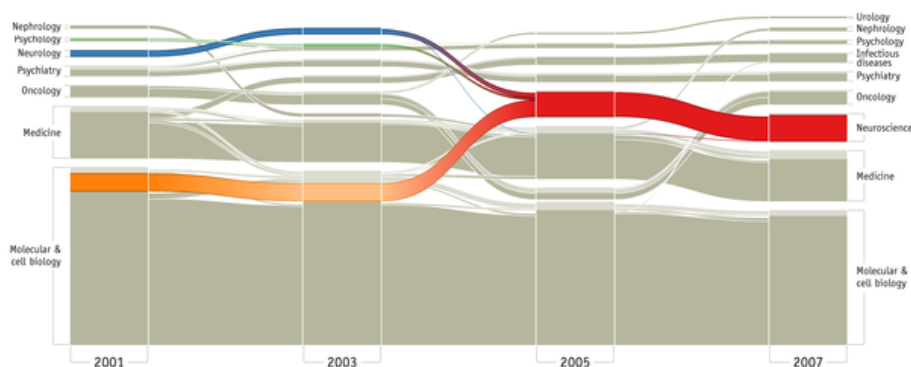


FIGURE 4.10 – Illustration des diagrammes alluviaux. Sur ce type de restitutions, les caractéristiques macroscopiques (par exemple la fusion de deux communautés) sont facilement identifiables. Image provenant de [171].

4.3.1.3 Séquences de graphes

La restitution de tracés représentant l'état du graphe à différents pas de temps a conduit à plusieurs propositions (par exemple dans [172] et [131]).

Ces approches sont particulièrement adaptées pour la comparaison de petits graphes à différents pas de temps mais elles comportent un inconvénient majeur identifié notamment par [156] : des changements radicaux dans la structure du graphe entre deux pas de temps entraînent une importante discontinuité dans sa représentation graphique. Le tracé correspondant au pas de temps t peut ainsi être totalement différent du tracé au pas de temps $t - 1$, et la carte mentale correspondante - représentation mentale du graphe - peut être perturbée [137, 150]. Ceci peut causer une charge cognitive additionnelle qui peut être nuisible au processus de visualisation. Pour tenter de pallier ce problème, différents mécanismes ont été proposés pour préserver la carte mentale en assurant la stabilité du graphe [62].

4.3.1.4 Graphes animés

Un graphe animé semble être une alternative naturelle pour représenter un graphe dynamique associé à un processus évoluant dans le temps [135]. Très représentés dans la littérature [59, 139, 17, 111, 122], les graphes animés peuvent être définis comme une animation automatique ou interactive du tracé d'une séquence de graphes. D'après certaines études [20, 87], l'œil humain est particulièrement bien adapté pour la détection et la classification du mouvement. L'animation d'une séquence de graphes peut donc être un outil bénéfique pour la bonne compréhension des dynamiques de l'évolution. De plus, cette approche tend à préserver la carte mentale de l'utilisateur puisque les discontinuités entre deux pas de temps successifs (par exemple le changement de position d'un nœud de (x_1, y_1) à (x_2, y_2)) disparaissent via l'ajout de transitions (déplacement du nœud de (x_1, y_1) vers (x_2, y_2)). Idéalement, les animations sont locales et ne concernent que les parties du graphe qui évoluent entre deux pas de temps [2]. En sus, certains mécanismes permettent d'attirer l'attention de l'utilisateur vers ces évolutions comme par exemple un nœud devenant transparent avant de totalement disparaître [44, 7].

Nous pouvons distinguer deux approches permettant d'aborder les graphes animés : l'approche *Online* et l'approche *Offline*. L'approche *Online* met à jour le tracé à chaque pas de temps en ne considérant que le pas de temps précédent tandis que l'approche *Offline* considère le graphe dans son ensemble. Dans les deux cas, des animations sont calculées afin d'assurer la transition fluide entre deux pas de temps. L'approche *Offline* est particulièrement adaptée pour assurer la stabilité du graphe puisque la séquence dans son ensemble est connue, ce qui permet de pouvoir cal-

culer un tracé global (et un ensemble d’animations) minimisant les discontinuités (comme dans le système proposé par [53]). Frishman et al. proposent par exemple une méthode [77] qui limite les mouvements trop importants via l’introduction d’une pondération pour chacun des nœuds assurant qu’un nœud éloigné de tout changement ne change pas de position. Cependant, dans de nombreux cas comme la surveillance en temps réel, la séquence entière n’est pas disponible.

Plusieurs méthodes ont ainsi été proposées pour l’approche *Online*. Il est par exemple possible d’employer une version incrémentale des algorithmes de tracé de type force et ressorts [134]. Dans ce cas, le graphe est assimilable à une simulation physique exécutée en temps réel avec une restitution graphique. Ces algorithmes évitent les déplacements trop importants entre deux images consécutives en limitant la vitesse maximum régissant le mouvement des nœuds. Ces algorithmes *Online* sont le plus souvent sujets à des contraintes de restitution qui nécessitent des calculs intensifs et requièrent l’implémentation de mécanismes d’optimisation (accélération GPU par exemple) pour pouvoir être exécutés de manière fluide.

Notons toutefois que les travaux récents d’Archambault et al. [6] montrent que la préservation de la carte mentale n’est pas nécessairement une contrainte impérative. Ils comparent des graphes animés ayant une forte stabilité à une approche dite des “*small multiple*” (des graphes correspondant à différents pas de temps sont représentés dans une matrice arrangée de manière chronologique). Leurs résultats montrent que pour des tâches d’ordre topologique au sujet du graphe dynamique, l’approche *small multiple* obtient des performances équivalentes au graphe animé tout en étant plus rapide.

4.3.2 La dynamique dans *TempoSpring*

L’outil *TempoSpring*, étant dédié à un contexte de fouille visuelle exploratoire, ses utilisateurs n’ont pas ou très peu d’hypothèses *a priori* quant aux caractéristiques à découvrir dans les graphes visualisés. Le choix d’un pas de temps pertinent pour utiliser l’approche *small multiple* est donc un exercice délicat. Choisir un pas de temps trop élevé peut dissimuler des informations importantes (ex : un nœud qui apparaît puis disparaît dans le même intervalle) tandis qu’un pas de temps trop faible peut rendre le nombre de graphes affichés dans la matrice trop important. Pour ces raisons, nous avons adopté l’approche des graphes animés.

Plus précisément, nous avons choisi une stratégie “online” permettant de mettre à jour le graphe visualisé en fonction des événements qui interviennent. Celle-ci s’implémente facilement avec un algorithme de tracé force et ressort itératif. Il s’agit de maintenir une liste globale des nœuds et des liens présents et de la mettre à jour en fonction des événements qui apparaissent. Lorsqu’un nœud apparaît, il est ajouté à une position aléatoire et intégré à l’algorithme de tracé ; lorsqu’un nœud disparaît,

il est supprimé et n'intervient plus dans l'algorithme de tracé. Ces changements locaux influencent la position des nœuds auxquels ils sont reliés, sans avoir d'influence trop importante sur le reste du graphe. Ceci vise à limiter les perturbations de la carte mentale de l'utilisateur. Un seuil pour l'accélération des nœuds (quand les forces auxquelles ils sont sujets entraînent une importante vélocité) limite également les perturbations en permettant à l'utilisateur de suivre les changements de position. Une illustration de la dynamique dans *TempoSpring* est présentée sur la figure 4.11.

TempoSpring peut prendre en compte deux sources de données de graphes dynamiques. La première source consiste en un fichier statique représentant un graphe dynamique où les temps d'apparition et de disparition de chacun des nœuds sont spécifiés (comme nous l'avons vu ci-dessus avec la figure 4.2). Pour le cas d'un nœud qui apparaît à différents pas de temps, il suffit de le dupliquer dans le fichier de données. Un curseur "temporel" est alors présenté dans la restitution pour pouvoir explorer la ligne de temps du graphe. La deuxième source de données prise en charge nécessite de connecter *TempoSpring* à une source de données compatible qui indique, en temps réel, les apparitions et disparitions de nœuds et de liens. En accord avec ces événements, la liste de nœuds et de liens de *TempoSpring* est mise à jour et ceux-ci sont donc ajoutés ou retirés de la restitution graphique.

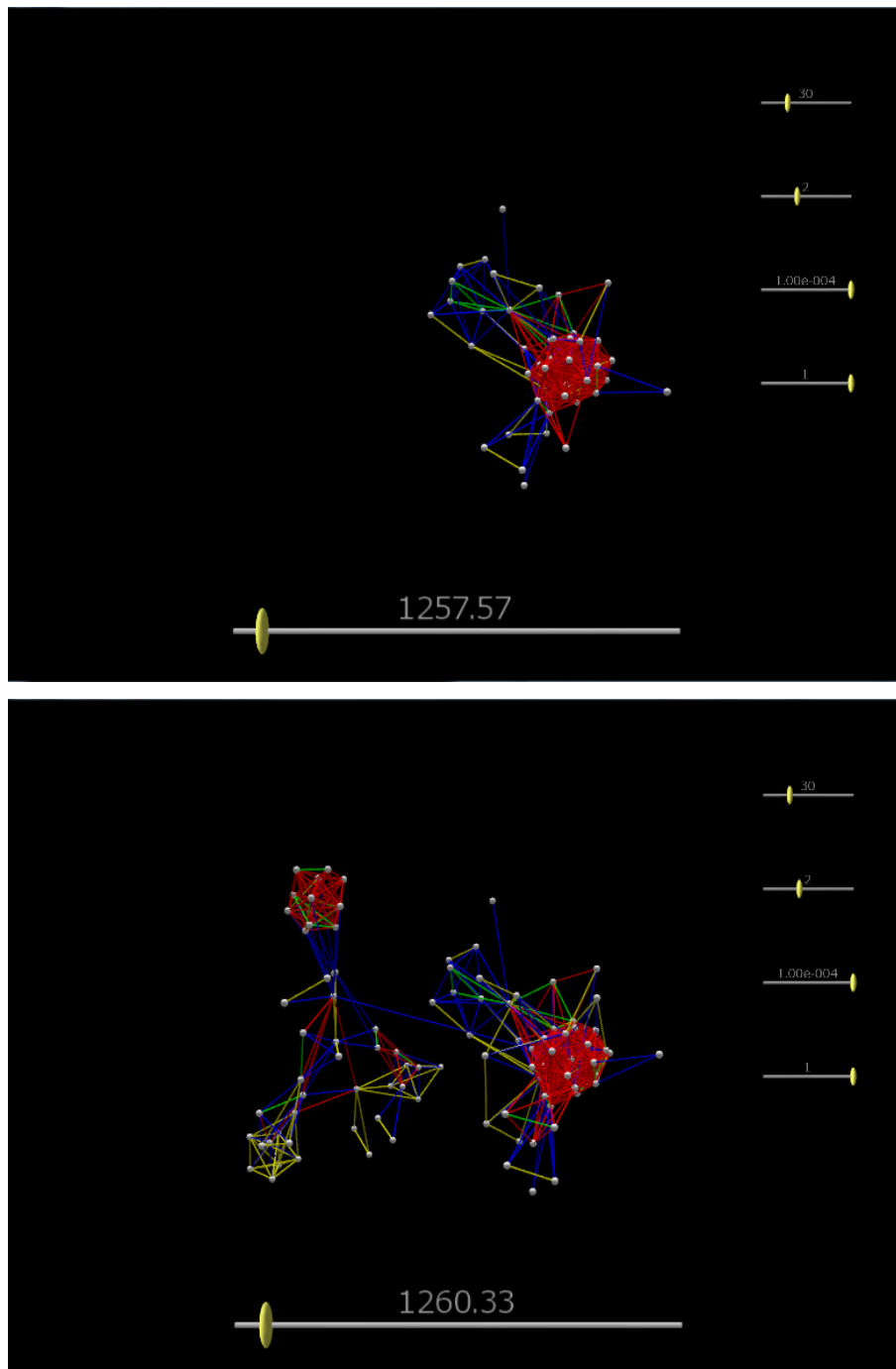


FIGURE 4.11 – Illustration de la dynamique dans *TempoSpring*. Sur les deux captures, la ligne de temps représente des années. Sur la première image, on peut observer l'état du graphe en 1257 et sur la seconde, son état en 1260. Malgré l'apparition de nombreux éléments (la structure en communauté située à gauche de la capture), le tracé des structures qui existaient déjà en 1257 n'a presque pas été modifié.

Détection de communautés en 3D

On ne peut se passer d'une méthode pour se mettre en quête de la vérité des choses.

—René Descartes

Sommaire

Introduction	76
5.1 Hypothèses	76
5.2 Protocole Expérimental	77
5.2.1 Matériel	78
5.2.2 Base de Données de Graphes	79
5.2.3 Participants	79
5.2.4 Procédure Expérimentale	81
5.3 Résultats	82
5.3.1 Qualité de la détection de communautés	82
5.3.2 Temps de Réponse	83
5.3.3 Perception des Participants	83
5.3.4 Discussion	85
5.4 Deuxième expérience	86
5.4.1 Améliorations du protocole expérimental	86
5.4.2 Résultats	87
Discussion	88

Introduction

La principale innovation de l’outil *TempoSpring* présenté dans le chapitre précédent est la possibilité de restituer les tracés de graphes en 2D, 3D monoscopique et 3D stéréoscopique. Comme nous l’avons vu dans le chapitre 2, la 3D stéréoscopique semble apporter un gain de performance dans différentes tâches de visualisation. En visualisation de graphes, elle a été validée à travers différentes études (voir section 2.3.1) pour des tâches locales comme la détection de chaînes. Mais comme nous l’avons rappelé dans la section 3.1.2, en fouille visuelle de graphes, les structures recherchées sont bien souvent d’ordre macroscopique, comme par exemple des communautés d’individus. Dans un monde de plus en plus connecté, l’identification de ces communautés est au centre de nombreux domaines d’application comme les recommandations en ligne [164] ou la configuration de réseaux sans fil *ad hoc* [39]. Pour considérer l’utilisation de la 3D stéréoscopique dans un système de fouille visuelle de graphes, il nous a donc paru primordial d’évaluer au préalable son apport pour une tâche de détection de communautés.

Dans ce chapitre, nous décrivons l’étude expérimentale que nous avons menée pour étudier l’impact de la 3D stéréoscopique pour une tâche de détection de communautés par rapport à une restitution 3D monoscopique et à une restitution 2D. Dans la première section nous énumérons les hypothèses que nous avons formulées à partir des travaux de la littérature. Dans la deuxième section nous décrivons les différents points de notre protocole expérimental : le matériel utilisé, la base de données de graphes que nous avons employée, les participants qui ont pris part à l’étude et la procédure expérimentale que nous avons suivie. Dans la troisième section nous présentons les résultats obtenus en terme de qualité de la détection, de temps de réponse et de préférence des participants. Nous discutons également de la présence de certains biais qui *a posteriori* nous ont paru pouvoir découler de certains choix expérimentaux. Pour affiner et compléter les résultats obtenus, nous avons effectué une deuxième expérimentation que nous exposons dans la quatrième section. Nous y décrivons les changements apportés au protocole expérimental et les résultats obtenus.

5.1 Hypothèses

Dans l’expérience, les participants doivent analyser visuellement un graphe et identifier le nombre de communautés qui le composent. Ces communautés ont des niveaux de chevauchements de taille variables déterminés par la complexité *a priori* du graphe. Pour la condition 2D, l’intégralité du graphe est visible dès son affichage. Pour les conditions 3D mono et 3D stéréo, des interactions peuvent être nécessaires pour révéler des parties sujettes à l’occlusion. Dans ce cadre, les hypothèses testées

sont les suivantes :

1. Quelle que soit la condition expérimentale, le nombre d'erreurs dans la détection croît avec la complexité des graphes visualisés.
2. La condition 2D ne requérant aucune interaction pour révéler l'intégralité du graphe, ses temps de réponse sont inférieurs en moyenne aux autres conditions.
3. La condition 3D stéréo améliorant la perception de la profondeur par rapport à la condition 3D mono, ses performances sont plus élevées.
4. La 3D stéréo peut dérouter les participants moins familiers avec cette technologie et requiert un temps d'adaptation augmentant les temps de réponse.
5. A l'instar des résultats de la littérature, la 3D stéréo permet d'interpréter des graphes plus complexes et obtient donc moins d'erreurs pour cette catégorie de graphes.
6. Pour les graphes les plus simples, la condition 2D obtient de meilleures performances.

5.2 Protocole Expérimental

Les trois conditions de visualisation suivantes ont été évaluées durant cette expérience. Elles reposent sur trois projections de tracés de graphes calculés avec l'algorithme de Fruchterman-Reingold [78], qui est applicable à des espaces 2D et 3D.

- 3D Monoscopique (3D mono) : un tracé 3D est pré-calculé en utilisant la version tridimensionnelle de l'algorithme de Fruchterman-Reingold et affiché sur une surface 2D via une projection en perspective linéaire. En plus du (dé)zoom contrôlé par la molette de la souris, les participants ont la possibilité de faire tourner leur point de vue autour du graphe (rotation sur x et y) via la technique "d'arcball" [183]. Cette technique garantit le maintien du graphe dans le champ de vision du participant, contrairement aux techniques d'exploration libre. De plus, certaines études suggèrent que les mouvements de rotation sur un objet transmettent un meilleur ressenti de la profondeur que les mouvements de translations [46].
- 3D Stereoscopique (3D stereo) : le même tracé et la même projection en perspective linéaire que pour le cas précédent sont utilisés mais avec l'introduction d'un deuxième point de vue -calculé en temps réel- permettant d'introduire la disparité binoculaire : un point de vue pour chaque œil avec un léger décalage le long de l'axe horizontal afin d'imiter la séparation naturelle entre les yeux humains. Les mêmes interactions que pour la 3D mono sont disponibles.

- 2D : un tracé 2D est pré-calculé en utilisant l’algorithme de Fruchterman-Reingold et affiché sur une surface 2D. Les participants ont la possibilité de (dé)zoomer. Afin d’assurer la cohérence de l’usage de la souris tout au long de l’expérience, une rotation horaire ou anti-horaire du graphe sur l’axe optique est proposée.

Notre choix d’algorithme de placement s’est porté sur Fruchterman-Reingold et non sur un modèle énergétique plus sophistiqué comme l’algorithme de LinLog [147] pour la raison suivante. Les représentations issues de ce dernier tendent à regrouper les communautés dans des amas très denses et à renforcer leur écartement. Cette représentation soulève des questions dans la prise en compte des chevauchements, et nous a paru moins adaptée pour la comparaison des interprétations visuelles des graphes dans nos différentes conditions expérimentales. Mais il serait intéressant de pouvoir rebâtir un protocole expérimental avec cet algorithme.

Le tableau 5.1 récapitule les mécanismes de perception de la profondeur mis en œuvre dans les conditions monoscopique et stéréoscopique.

Mécanisme de Perception	Implémentations
Disparité binoculaire	Deux vues parallèles séparées horizontalement + lunettes stéréoscopiques actives
Mouvements	Rotation du graphe
Tailles relatives*	Projection en perspective linéaire (OpenGL)
Occlusion	Restitution 3D et tracé de Fruchterman permettant les occlusions

TABLE 5.1: Implémentation des différents mécanismes de perception. La seule différence entre les deux conditions expérimentales provient de l’addition de la disparité binoculaire pour la condition stéréoscopique. (*) Un algorithme d’anticrénelage ($16\times$) améliore la qualité de ces mécanismes.

5.2.1 Matériel

Le système de visualisation utilisé est exécuté sur un ordinateur doté d’un processeur Intel Core 2 Duo (3.00 Ghz) E8400 avec 4 GB de RAM et une carte graphique Nvidia Quadro FX 3800. Tous les graphes sont affichés en nuances de gris sur un fond noir, avec un algorithme d’anticrénelage afin d’améliorer la qualité de la restitution. La visualisation est projetée sur un mur blanc via un projecteur 3D Acer H5360 (écran de $2,30 \times 1,30 \text{ m}^2$) ayant une résolution de 1280×720 pixels (angle de vue de 0.05 degrés pour un pixel au centre de l’écran). Notre système utilise une technique de stéréoscopie dite active avec des lunettes à obturateur Nvidia 3D Vision. Utiliser ces lunettes entraîne une baisse significative de la luminosité perçue. Ainsi, afin d’éviter tout biais lié au port de ces lunettes, les participants doivent les garder durant toute l’expérience, quelle que soit la méthode de visualisation utili-

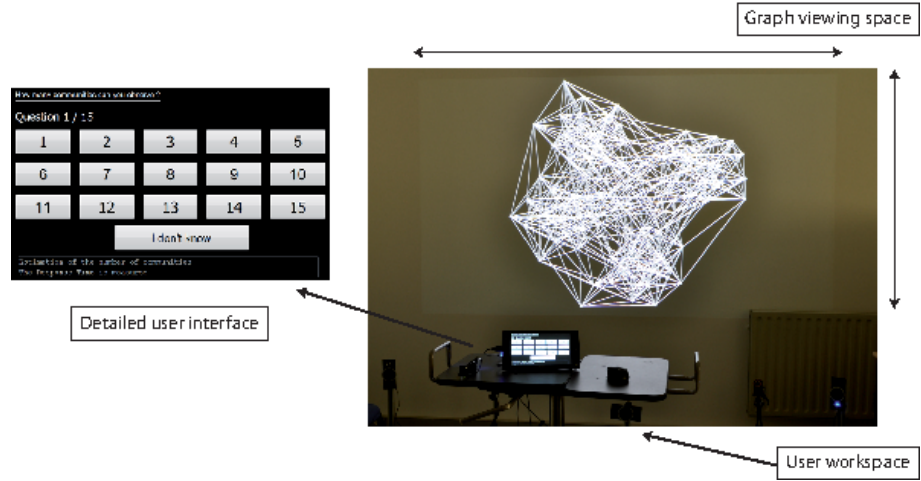


FIGURE 5.1 – Photographie de l'installation et une capture d'écran de l'interface utilisée par les participants.

sée. Les participants peuvent interagir avec le système via une souris sans fil. Pour faciliter la saisie et minimiser les perturbations sur la tâche visuelle, les réponses sont saisies par les participants en utilisant une tablette tactile : différents numéros (allant de 1 à 15, plus une option "Je ne sais pas") sont affichés et l'utilisateur clique sur le numéro correspondant à sa réponse (figure 5.1).

5.2.2 Base de Données de Graphes

La comparaison entre les différentes conditions est faite sur une classe de graphes pseudo-aléatoires comportant une structuration non triviale en communautés. Ils sont obtenus avec un modèle de génération classique $G(k; nv; p_{int}; p_{ext})$ qui dépend de 4 paramètres (c.f. : [45, 82]) : le nombre k de communautés *a priori*, le nombre nv de nœuds par communautés, la probabilité p_{int} (resp. p_{ext}) d'avoir un lien entre deux nœuds d'une même communauté (resp. différentes communautés). Nous avons ainsi produit 545 graphes avec les paramètres suivants : $k \in \{4, 5, \dots, 11\}$, $nv = 10, 20, 30, 40$, $p_{int} = 0.5, 0.6, 0.7, 0.8$ et $p_{ext} = 0.02, 0.03, 0.04, 0.05, 0.065, 0.07, 0.08, 0.1$. Les paramètres p_{int} et p_{ext} ont été déterminés de manière empirique lors d'une étude préalable menée par deux utilisateurs confirmés. Notons que les paramètres p_{int} et p_{ext} ont été considérés par paires formant ainsi les ratios de complexité : $\frac{p_{int}}{p_{ext}} = \left\{ \frac{0.02}{0.8}, \frac{0.02}{0.7}, \frac{0.03}{0.8}, \frac{0.03}{0.7}, \frac{0.03}{0.6}, \frac{0.04}{0.7}, \frac{0.03}{0.5}, \frac{0.05}{0.8}, \frac{0.05}{0.7}, \frac{0.05}{0.6}, \frac{0.065}{0.6}, \frac{0.07}{0.6}, \frac{0.1}{0.8}, \frac{0.08}{0.6} \right\}$ Trois exemples de graphes 2D générés par ce modèle sont présentés sur la figure 5.2.

5.2.3 Participants

35 participants (25 hommes et 10 femmes) ont pris part à cette expérience. Agés de 20 à 50 ans, 30 d'entre eux venaient d'un cursus informatique (chercheurs et étudiants). Trois des participants étaient gauchers et utilisaient la souris de la main

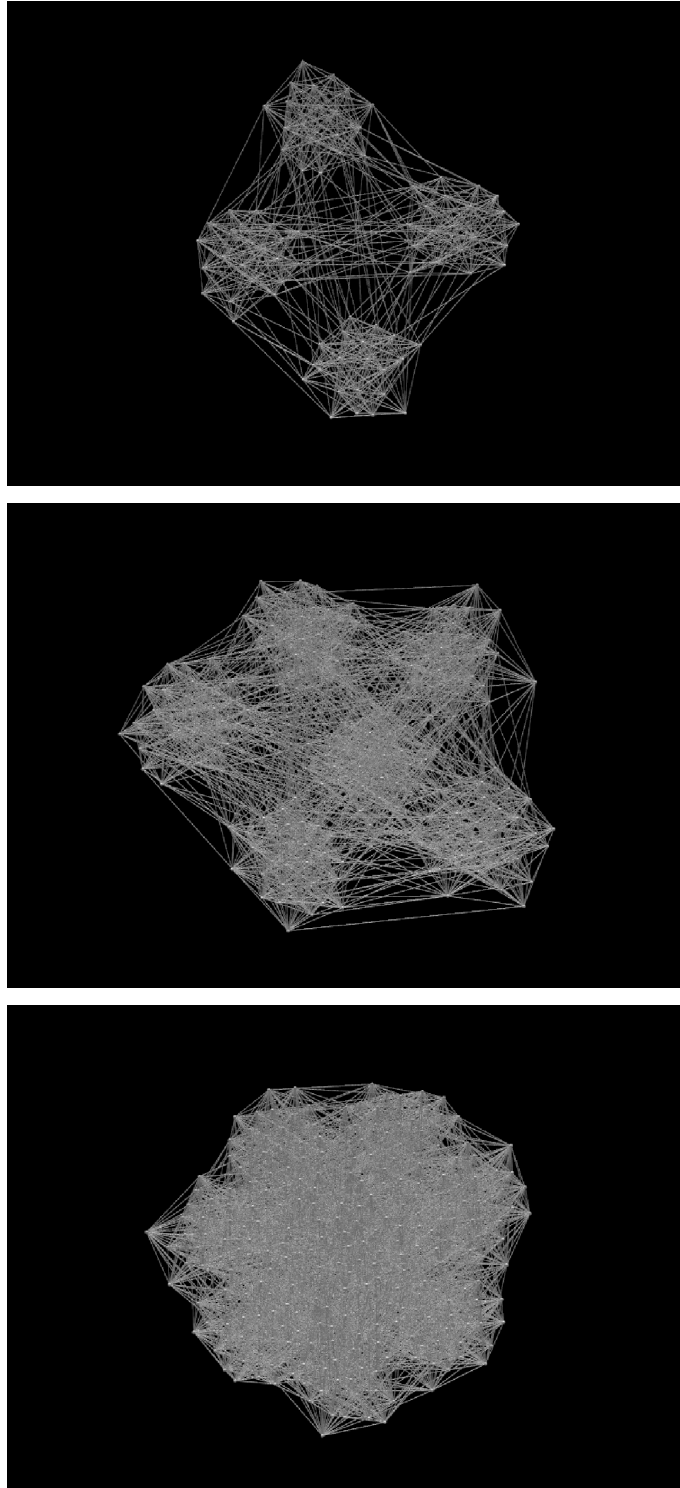


FIGURE 5.2 – Captures de tracés de graphes de complexité croissante. $G_1 = G(k = 4, nv = 20, p_{ext} = 0.05, p_{int} = 0.8)$; $G_2 = G(k = 6, nv = 30, p_{ext} = 0.05, p_{int} = 0.7)$; $G_3 = G(k = 8, nv = 30, p_{ext} = 0.07, p_{int} = 0.8)$

droite. Seulement deux des participants n’avaient jamais visualisé la moindre image stéréoscopique et 11 parmi les 35 n’étaient pas familiers des logiciels 3D comme les jeux vidéos.

5.2.4 Procédure Expérimentale

Afin de limiter la durée de l’expérience, 15 tracés successifs par condition de visualisation étaient présentés à chaque participant. Ces tracés étaient tirés de manière aléatoire (sans duplication dans une même condition) depuis la base de données. Afin d’éviter tout biais d’apprentissage potentiel, les 3 conditions de visualisation apparaissaient dans un ordre aléatoire (par exemple : 15 tracés en 3D stéréo suivis de 15 en 3D mono suivis de 15 en 2D). La durée moyenne d’une session était de 15 minutes (28 au maximum). Cette expérience a engendré 1575 visualisations de graphes ($3 \text{ conditions} \times 15 \text{ graphes} \times 35 \text{ participants}$).

Avant le début de chaque session, des questions simples étaient posées aux participants afin de recueillir leur degré de familiarité avec la théorie des graphes, la visualisation de graphes, les affichages stéréoscopiques et l’utilisation de logiciels 3D. Une fiche de présentation expliquant la procédure à suivre était distribuée et une démonstration rapide du système était effectuée. Cette démonstration présentait le système de visualisation (les interactions disponibles, la tâche à effectuer) ainsi que les trois conditions de visualisation sur un graphe facilement appréhendable (3 communautés, 20 nœuds par communauté une importante probabilité p_{int} et une faible probabilité p_{ext}). A la suite de la démonstration, les participants devaient compléter une session d’entraînement afin de se familiariser avec le système. Cet entraînement comprenait 3 tracés de complexité croissante par condition de visualisation.

Il était ainsi demandé aux participants d’estimer le nombre de communautés présentes dans les graphes affichés le plus rapidement possible sans l’aide de l’expérimentateur. Si un participant n’arrivait pas à discerner les différentes communautés, la consigne était de passer au tracé suivant via la sélection du bouton “Je ne sais pas”. Une fois tous les tracés d’une condition de visualisation analysés, le participant passait à la condition suivante. L’expérimentateur accompagnait cette transition (“Vous avez terminé la condition XX, vous vous apprêtez à commencer la condition YY”). A la fin de la session, quelques questions étaient de nouveau posées aux participants pour connaître leurs préférences (condition la plus facile et la plus difficile) et l’auto-évaluation de leur performance (conditions avec les meilleures et moins bonnes performances).

5.3 Résultats

Dans cette section nous présentons les résultats de l'expérience selon trois critères : la qualité de la détection de communautés, le temps de réponse et la préférence des participants. Dans la suite nous notons I l'ensemble des instances i présentées aux participants avec le même ratio p_{ext}/p_{int} , $k_{i,ans}$ le nombre de communautés proposé par un participant pour une instance i et k_i le nombre *a priori* de communautés dans le modèle.

5.3.1 Qualité de la détection de communautés

Pour chaque condition de visualisation vm , l'erreur est mesurée par la moyenne des différences entre $k_{i,ans}$ et k_i :

$$Error_{vm}(I) = \frac{1}{card(I)} \sum_{i \in I} (|k_{i,ans} - k_i|) \quad (5.1)$$

Notons que la réponse “Je ne sais pas” était comptée comme une erreur (c'est à dire la détection d'une seule communauté), et que pour les valeurs les plus grandes de p_{ext} et les plus faibles de p_{int} , le terme “erreur” doit être nuancé car la détection peut être très ambiguë. Mais dans ces cas, l'évaluation de la qualité de la détection de communautés reste une question délicate. Notons que des travaux récents (e.g. [74]) ont permis d'améliorer l'évaluation avec des mesures de type modularité mais cette question dépasse largement le cadre de cette thèse.

Les résultats dépendent de la complexité de la structure du graphe mesurée ici par le ratio p_{ext}/p_{int} : pour de petites valeurs, les communautés sont facilement identifiables tandis que pour de plus grandes valeurs, d'importants chevauchements et des densités de communauté faibles rendent la détection délicate. Le tableau 5.2 montre que la 2D est significativement meilleure pour les graphes de faible complexité (Anova2 : $p_{value} = 0.01$) : les communautés sont bien séparées sur le tracé et facilement détectables sur un plan. Pour les graphes ayant une complexité plus importante, la stéréoscopie est légèrement plus performante (Anova2 $p_{value} = 0.1$). Mais cette différence n'est significative que pour $k > 7$ et pour des ratio de complexité supérieurs à 0.06 (Anova2 : $p_{value} = 0.02$), tel que nous le montre le tableau 5.3. La dimension additionnelle perçue combinée au mouvement semble aider à faire la distinction entre les différents agrégats et ce, même en présence de “bruit” (chevauchements). La situation est différente pour la 3D monoscopique, pour laquelle les occlusions expliquent certainement les différences avec la stéréoscopie. Quelle que soit la condition, la variation de l'erreur devient vite très importante dès que la complexité croît (tableau 5.4). Nous avons observé que cette variation est indépendante de k et qu'elle ne peut pas, dans notre échantillon de participants,

Complexité (p_{ext}/p_{int})	2D	3D mono	3D stereo
[0.02; 0.04]	0.10	0.37	0.27
]0.04; 0.06]	1.62	1.64	1.40
]0.06; 0.11]	3.27	3.22	2.78
]0.11; 0.15]	3.47	3.71	2.99

TABLE 5.2: Erreur dans la détection de communautés. Pour chaque méthode de visualisation vm et pour chaque intervalle de complexité (p_{ext}/p_{int}) I , l'erreur moyenne $Error_{vm}(I)$

être expliquée par l'absence de familiarité avec les logiciels 3D. Cependant des expériences complémentaires sont nécessaires avant de rejeter totalement cette hypothèse.

5.3.2 Temps de Réponse

Pour chaque méthode de visualisation, le temps de réponse $Time_{vm}(I)$ est la moyenne des temps de réponse des participants confrontés au même ratio p_{ext}/p_{int} . Le tableau 5.5 nous montre que le temps de réponse de la 2D est significativement plus faible que pour la 3D (Anova2 : $p_{value} \leq 0.001$), quelle que soit la complexité des graphes. De plus, les temps de réponses de la 3D monoscopique et de la 3D stéréoscopique sont très similaires. Cette différence entre 2D et 3D est probablement due à la nécessité d'interagir avec les graphes en 3D afin de les analyser dans leur ensemble.

5.3.3 Perception des Participants

Le tableau 5.6 indique que les participants semblent préférer la stéréoscopie et semblent l'avoir trouvée plus facile à appréhender que les autres conditions de visualisation. Pour la question "Quelle condition avez-vous trouvé la plus facile?", un test de Friedman [76] révèle que les résultats sont significatifs ($\chi^2(2) = 6.6$, $p_{value} < 0.05$). Pour la question "Avec quelle condition de visualisation pensez-vous avoir obtenu les meilleures performances?", un test de Friedman révèle que les résultats sont à nouveau significatifs ($\chi^2(2) = 5.7$, $p_{value} < 0.05$). Cependant, nous sommes conscients qu'un biais peut exister. En effet, l'expérience en elle-même illustre notre intérêt pour la 3D stéréoscopique, et les participants pouvaient être enclins à partager notre enthousiasme. Quoi qu'il en soit, une part de cette subjectivité est corroborée par les résultats de l'expérience. En effet, parmi les participants ayant indiqué avoir mieux réussi en 3D stéréo, 54% ont vu leur intuition confirmée par les résultats (tandis que seulement 15.5% ont obtenu leurs meilleurs

Complexité (p_{ext}/p_{int})	Méthode	$k = 4$	$k = 5$	$k = 6$	$k = 7$
[0.02; 0.04]	2D	0.0	0.0	0.1	0.05
	3D mono	0.0	0.0	0.13	0.08
	3D stereo	0.0	0.0	0.0	0.07
]0.04; 0.06]	2D	0.0	0.14	0.0	0.19
	3D mono	0.0	0.0	0.08	0.58
	3D stereo	0.0	0.0	0.13	0.9
]0.06; 0.11]	2D	0.0	0.0	0.5	1.35
	3D mono	0.0	0.13	0.32	1.07
	3D stereo	0.0	0.0	0.21	1.6
]0.11; 0.15]	2D	0.28	0.88	2	4.69
	3D mono	0.73	1.23	2.3	5.81
	3D stereo	0.93	1.35	2.06	6
		$k = 8$	$k = 9$	$k = 10$	$k = 11$
[0.02; 0.04]	2D	0.06	0.16	0.13	0.67
	3D mono	0.25	0.35	0.37	1.56
	3D stereo	0.16	0.11	0.42	1.67
]0.04; 0.06]	2D	1.27	2.65	3.35	6.94
	3D mono	1.45	1.92	4.13	5.93
	3D stereo	0.63	0.69	3.81	5.5
]0.06; 0.11]	2D	2.24	3.95	6.15	7.9
	3D mono	1.63	4.44	6.12	7.61
	3D stereo	1.9	3.42	5	6.88
]0.11; 0.15]	2D	6.14	7.13	9	10
	3D mono	5.44	6	8.75	8.75
	3D stereo	5.25	8	6.6	6

TABLE 5.3: Erreur dans la détection de communautés. Pour chaque méthode de visualisation vm et pour chaque intervalle de complexité (p_{ext}/p_{int}) I , l'erreur moyenne $Error_{vm}(I)$ en fonction de k

Complexity (p_{ext}/p_{int})	2D	3D mono	3D stereo
< 0.06	1.28	1.16	1.67
≥ 0.06	3.63	3.69	3.38

TABLE 5.4: Ecart type de l'erreur dans la détection de communautés en fonction de la complexité (p_{ext}/p_{int})

Complexité (p_{ext}/p_{int})	2D	3D mono	3D stereo
[0.02; 0.04]	7.3	14.2	12.3
]0.04; 0.06]	11.1	17.9	17.7
]0.06; 0.11]	12.4	22.5	24.7
]0.11; 0.15]	13.1	21.9	21.1

TABLE 5.5: Temps de réponse moyen $Time_{vm}(I)$ en secondes, pour chaque méthode de visualisation vm et pour chaque intervalle de complexité (p_{ext}/p_{int})

Réponse	2D	3D mono	3D stereo	NSP
Plus facile	14.2	0	68.6	17.2
Plus difficile	37.1	43	5.7	14.2
Meilleures performances	11.3	0	74.3	14.4
Moins bonnes performances	43	34.3	5.7	17

TABLE 5.6: Perception subjective des participants. Pour chaque méthode de visualisation vm , pourcentage des participants ayant désigné que vm comme étant la plus facile (resp. le plus difficile) et celle avec laquelle ils ont estimé avoir obtenu leur meilleure (resp. moins bonne) performance.

résultats avec la 3D monoscopique).

5.3.4 Discussion

Les résultats de cette première expérience sont encourageants et semblent confirmer plusieurs de nos hypothèses initiales. Tout d’abord, la difficulté de la tâche croît vraisemblablement avec la complexité des graphes visualisés (hypothèse 1). Devant cette complexité croissante, les trois conditions étudiées ne sont pas égales : la condition 2D obtient les meilleurs résultats pour les complexités faibles tandis que la 3D stéréoscopique prévaut pour les complexités plus importantes. Ce qui semble indiquer l’existence d’un effet seuil, et semble confirmer les hypothèses 5 et 6. Conformément à l’hypothèse 3, la 3D monoscopique obtient globalement de moins bons résultats que la 3D stéréoscopique, mais ne semble pas obtenir de temps de réponse plus rapides, ce qui invalide l’hypothèse 4. Notons cependant que certains participants ont déclaré avoir eu besoin d’un petit temps d’adaptation lors de la série d’entraînements avant de se familiariser avec la stéréoscopie. Enfin, les temps de réponse sont toujours plus faibles pour la condition 2D (hypothèse 2).

Cette expérience nous a également permis de mettre en évidence certains biais dans notre protocole expérimental. Tout d’abord, nous avons sous-estimé les capacités des participants en proposant certains graphes trop simples qui représentent peu d’intérêt d’un point de vue fouille visuelle. A l’inverse, certains graphes se

sont révélés être extrêmement durs à analyser. Nous avons également constaté que les paramètres p_{ext} et p_{int} n'affectent pas la difficulté observée de la même façon. A valeurs égales, deux ratios de complexités p_{ext}/p_{int} peuvent engendrer des erreurs mesurées différentes. De plus, nous avons accordé beaucoup d'importance à des observations effectuées dans une étude préalable nous avertissant d'un possible biais d'apprentissage lorsque les graphes visualisés sont identiques pour les trois conditions. Ces observations nous ont conduits à assigner, pour chaque participant et chaque condition de visualisation, des séries de graphes aléatoires différentes. En sus de la variabilité des deux autres paramètres k et nv , tous ces facteurs ont engendré des données difficiles à analyser. Nous avons donc été conduits à procéder à une analyse statistique multivariée (Anova2 parfois aussi appelée Manova) qui est, dans notre cas, d'une puissance statistique contestable.

Notons toutefois qu'un test de Friedman a été calculé sur un échantillon comprenant uniquement les tracés visualisés sous les trois conditions. Portant sur 136 tracés, il valide nos résultats mais demande confirmation.

5.4 Deuxième expérience

Dans cette section nous présentons une deuxième série d'expériences visant à confirmer les résultats de la première expérience tout en rectifiant certains biais avec un nouveau protocole expérimental.

5.4.1 Améliorations du protocole expérimental

Dans cette expérience, nous avons cherché à approfondir la compréhension des résultats précédents en construisant un nouveau protocole qui contient une nouvelle base de données de graphes testés et une nouvelle procédure expérimentale.

Base de données de graphes Nous avons engendré les tracés de 135 graphes correspondant à toutes les combinaisons des paramètres suivants : $k \in [4, \dots, 12]$, $nv = 30$, $p_{int} = \{0.6, 0.7, 0.8\}$ et $p_{ext} \in [0.04, 0.05, \dots, 0.08]$. Ces paramètres engendrent une classe de graphes avec une forte probabilité de structuration en communautés qui ne sont pas directement apparentes. Le paramètre nv est gardé constant ($nv = 30$) pour s'assurer que la difficulté de la tâche ne dépende que du ratio de complexité p_{ext}/p_{int} (facteur de chevauchement) et du nombre de communautés k . Les 135 tracés sont aléatoirement répartis en 9 séquences distinctes s_i , $i = [1..9]$ de 15 tracés.

Procédure expérimentale Le processus d'assignation des tracés de graphes a été modifié par rapport à la première série d'expériences. Pour chaque participant, 15

tracés étaient alors tirés aléatoirement et indépendamment pour chaque condition.

Cette fois-ci, une séquence de 15 tracés a été affectée à chaque participant, et chaque séquence a été vue par deux participants sous les 3 conditions. L'ordre de visualisation des graphes à l'intérieur de chaque séquence est tiré aléatoirement pour chaque condition. Ainsi, $9 \times 2 = 18$ participants ont pris part à l'expérience et $135 \times 2 = 270$ tracés ont été présentés sous les 3 conditions. Les participants (3 femmes et 15 hommes âgés de 20 à 44 ans) étaient tous des étudiants ou chercheurs en informatique.

Afin d'éviter un biais lié à l'ordre des conditions, pour chaque séquence, le premier participant commençait par une visualisation de la séquence en 3D stéréoscopique, puis en 3D monoscopique avec une inversion pour le deuxième participant concerné par la même séquence. Pour limiter les phénomènes d'apprentissage, la condition 2D était toujours présentée entre les deux autres conditions.

5.4.2 Résultats

Dans cette deuxième expérience, l'erreur dans la détection est calculée de la même manière que dans la première (éq. 5.1). Comme nous avons restreint les paramètres de génération des graphes, nous pouvons considérer dans cette étude toutes les observations. L'erreur moyenne sur l'ensemble des tracés pour la 3D monoscopique (resp. 3D stéréoscopique) est de $Error_{mono} = 2.5$ (resp. $Error_{stereo} = 1.8$). Pour la condition 2D, elle est de $Error_{2D} = 2.7$. Rappelons que dans cette expérience, nous ne considérons pas les plus petits graphes pour lesquels la 2D a obtenu de très bons résultats dans la première expérience ; ce qui explique certainement cette erreur moyenne élevée. Un test statistique de Kruskal-Wallis [120] ($\chi^2(2) = 18.96, p_{value} \leq 0.0001$) révèle qu'il existe une différence significative entre les conditions de visualisation. Une analyse *post-hoc* de comparaisons pair à pair avec correction de Bonferroni [26] révèle d'où provient cette différence : un test sur les rangs de Wilcoxon montre que la condition monoscopique est associée à une erreur plus importante que la condition stéréoscopique de manière significative ($Wilcoxon(269) = 24560$, avec $z_{value} = 4.16, p_{value} < 0.0001$). La condition 2D est également associée à une erreur moyenne plus importante que la condition stéréoscopique ($Wilcoxon(269) = 79770$, avec $z_{value} = 4, p_{value} < 0.0001$).

Concernant les temps de réponse, un test de Kruskal-Wallis révèle qu'il existe une différence significative entre les trois conditions de visualisation ($\chi^2(2) = 139.5, p_{value} \leq 0.0001$). Une analyse *post-hoc* de comparaisons pair à pair avec correction de Bonferroni révèle que cette différence provient de la condition 2D qui obtient globalement des temps de réponse plus rapides avec $T_{2D} = 10sec$ (2D vs 3D stéréo : $Wilcoxon(269) = 55062$, avec $z_{value} = -9.91, p_{value} < 0.0001$ et 2D vs 3D mono : $Wilcoxon(269) = 53949, z_{value} = -10.52, p_{value} < 0.0001$). Cepen-

dant, nous n’observons aucune différence significative entre les temps de réponse de la condition monoscopique et de la condition stéréoscopique : $T_{mono} = 17sec$ et $T_{stereo} = 17.3sec$ ($Wilcoxon(269) = -3098$, avec $z_{value} = -1.21$, et $p_{value} = 0.22$). Ces résultats semblent confirmer, dans l’ensemble, les résultats obtenus durant la première série d’expériences.

Discussion

A notre connaissance, ce chapitre présente une recherche pionnière sur l’utilisation de la stéréoscopie dans un problème de visualisation qui a connu un intérêt croissant durant la dernière décennie : la détection de communautés dans de grands graphes. Notre expérience révèle une différence importante entre la 3D stéréoscopique et la 3D monoscopique qui a été vivement critiquée par les communautés de Graph Drawing et de visualisation de réseaux sociaux. De plus, bien que le débat reste ouvert, nos résultats expérimentaux indiquent que la stéréoscopie peut prévaloir sur la 2D pour les graphes aux structures complexes comportant de nombreuses communautés de densités variables avec de nombreux chevauchements. Des études complémentaires sont toutefois nécessaires afin de confirmer ces résultats sur des populations plus larges et pour pouvoir comprendre les différences observées.



6

Etude des interactions pour la tâche de détection de communautés

There's real poetry in the real world. Science is the poetry of reality.

—Richard Dawkins

Sommaire

Introduction	90
6.1 Observations et hypothèses	90
6.2 Mesures classiques d'interactions	91
6.3 Indicateurs de mouvements	94
6.4 Analyse des Mouvements	97
6.4.1 Approche exploratoire : classification non supervisée	97
6.4.2 Analyse statistique des indicateurs	98
6.4.3 Discussion	100
6.5 Analyse des points de vue	101
6.5.1 Carte de chaleur 3D	101
6.5.2 Résultats	102
Discussion	103

Introduction

Nous avons vu dans le chapitre précédent que la 3D stéréoscopique semble être très prometteuse pour l'exploration visuelle de certaines structures dans des graphes. L'observation des enregistrements vidéo des expériences que nous avons menées nous a conduits à faire l'hypothèse suivante : en phase d'exploration, l'interaction avec les représentations visuelles des graphes diffère selon la présence ou l'absence de la disparité binoculaire. Pour essayer de caractériser cette différence nous avons mené une analyse en deux phases : (1) l'analyse des enregistrements des mouvements du pointeur de la souris utilisée pour les rotations des tracés de graphe, puis (2) l'analyse des zones des tracés perçues pendant l'exploration. Pour cette étude, nous nous sommes restreints aux tracés ayant nécessité un temps d'exploration supérieur à 3 secondes (207 tracés au total) qui sont potentiellement suffisamment complexes pour la tâche de détection de communautés pour engendrer de l'interaction. Les résultats obtenus semblent confirmer notre hypothèse.

Ce chapitre porte sur la description de l'étude des interactions enregistrées lors de la deuxième expérience utilisateur que nous avons évoquée dans la section 5.4. La première section liste les phénomènes que nous avons observés lors de la première expérience utilisateur décrite dans le chapitre précédent. Nous y énumérons également les hypothèses que ces observations nous ont menés à formuler. Dans la deuxième section nous rappelons les principaux travaux portant sur l'étude des interactions avec un périphérique de pointage dans la littérature. Cet état de l'art nous a permis de définir plusieurs indicateurs permettant de caractériser les interactions enregistrées. Nous les décrivons dans la troisième section. Dans la quatrième section, nous exposons l'analyse de ces indicateurs avec une approche exploratoire sous la forme d'une classification non supervisée puis avec une analyse statistique. Nous terminons ce chapitre en étudiant les interactions comme étant une suite de points de vue que nous analysons via la construction d'une carte de chaleur 3D qui propose une synthèse visuelle des temps passés sur les différents points de vue des utilisateurs.

6.1 Observations et hypothèses

Dans le contexte expérimental défini dans le chapitre 5, les interactions évaluées lors de nos expériences concernent l'utilisation de la souris. Ici, un mouvement de souris est directement associé à une rotation du tracé du graphe centrée sur son barycentre qui est positionné au centre de l'écran. L'utilisation de la molette de la souris permet de zoomer/dézoomer sur le tracé.

Lors de la première série d'expériences décrites dans le chapitre 5, des observations informelles ont été formulées sur ces interactions :

1. En stéréo, les utilisateurs semblent effectuer des pauses dans l'exploration du tracé (interruption du mouvement de la souris). Réciproquement, en mono, les utilisateurs ont tendance à déplacer la souris en permanence.
2. En mono, les mouvements de la souris ont tendance à être assez longs et continus. Réciproquement, en stéréo, les mouvements ont tendance à être plus courts avec des changements de direction fréquents.

Ces observations nous ont conduits à modifier notre premier protocole expérimental pour pouvoir analyser les interactions. Tout d'abord, nous avons ajouté deux caméras vidéo à notre installation afin de pouvoir visionner *a posteriori* les sessions expérimentales (figure 6.1). Nous avons aussi modifié notre système afin de permettre l'enregistrement des positions des points de vue adoptés par les utilisateurs (ceux-ci sont décrits plus en détail dans la section 6.5). Enfin, nous avons également modifié la séquence de graphes visualisés pour s'assurer que chaque participant analyse un graphe donné sous toutes les conditions expérimentales.

Dans ce cadre, les hypothèses testées sont les suivantes :

1. En stéréo et en mono, l'interaction se décompose en une succession de phases "d'exploration dynamique" caractérisées par des mouvements de souris, et des phases "d'exploration statique" caractérisées par l'absence de mouvements.
2. En stéréo, le temps total des phases d'exploration statiques que l'on peut également appeler "pauses" est proportionnellement plus grand qu'en mono.
3. En stéréo, les changements de direction de la souris sont plus fréquents.
4. En mono, il y a plus d'interactions avec le tracé (de mouvements) et la distance totale parcourue par la souris est plus élevée qu'en stéréo.

6.2 Mesures classiques d'interactions

Le cadre expérimental le plus exploré pour l'analyse des interactions avec une souris est certainement celui du déplacement du pointeur, selon une trajectoire linéaire, d'un point A vers une cible B de largeur w située à une distance $d(A, B)$ de A . La célèbre "loi" de Fitts [73] exprime le temps nécessaire pour cette tâche :

$$T_{Fitts} = \alpha + \beta \log \left(\frac{d(A, B)}{w} + c \right) \quad (6.1)$$

où α et β sont des constantes déterminées de manière empirique et c est une constante communément fixée à 1 afin de s'assurer que la partie logarithmique de l'équation, appelée indice de difficulté (ID), reste positive.

Des extensions ont été proposées pour des trajectoires plus complexes [1] et des

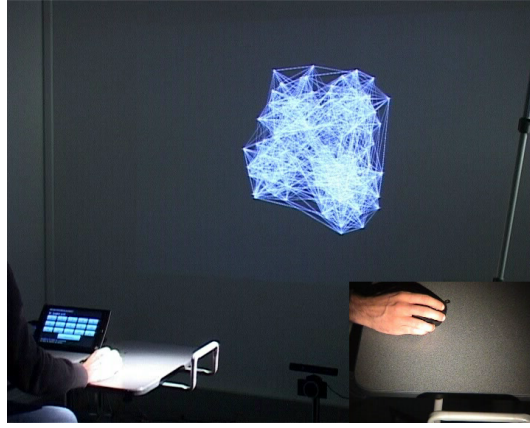


FIGURE 6.1 – Capture d’écran d’une vidéo de la deuxième expérimentation. Nous avons utilisé deux caméras lors de la deuxième expérience pour visionner *a posteriori* les interactions des participants. La première caméra située derrière eux enregistre la restitution graphique tandis que la deuxième enregistre les mouvements de la souris. Le visionnage de ces vidéos nous a aidé à formuler les 4 hypothèses testées.

cibles de tailles différentes [173]. En particulier, on peut alors prendre en compte les trois premières dérivées (x', y') , (x'', y'') , (x''', y''') de la position de la souris (x, y) qui s’interprètent respectivement comme la vitesse, l’accélération et la secousse le long des axes de déplacement horizontal et vertical. A ces informations, s’ajoute la courbure locale :

$$Courb = \frac{|x'y'' - x''y'|}{(x'^2 y'^2)^{\frac{3}{2}}} \quad (6.2)$$

qui permet d’exprimer la tendance locale pour une position (x, y) du mouvement entre une trajectoire linéaire et une trajectoire circulaire.

D’autres mesures ont été proposées pour quantifier l’écart entre une trajectoire donnée et une trajectoire optimale définie comme étant la ligne droite entre 2 points. Considérons une trajectoire optimale. Et notons δ_t la distance entre cette trajectoire et la position du pointeur à l’instant t et avec n le nombre total de positions. Différentes mesures ont été proposées comme notamment [133, 179] :

- Le décalage du mouvement est défini comme la moyenne des distances :

$$\bar{\delta} = \frac{1}{n} \sum_t \delta_t \quad (6.3)$$

- La variabilité du mouvement définie comme l’estimateur sans biais de l’écart type des distances de n positions par rapport à la moyenne :

$$\hat{V}_{\delta} = \sqrt{\frac{\sum \delta_t^2 - \bar{\delta}^2}{n - 1}} \quad (6.4)$$

- L'erreur du mouvement qui est la moyenne des distances absolues :

$$Err_\delta = \frac{1}{n} \sum_t |\delta_t| \quad (6.5)$$

- La mesure du nombre de croisements (notée TAC pour *Task Axis Crossings*) avec l'axe de la trajectoire optimale est définie comme le nombre de fois que le pointeur croise cet axe
- Deux mesures principales de changement de direction ont été proposées : MDC (*Movement Direction Change*) est le nombre de changements de direction du pointeur par rapport à l'axe optimal de la trajectoire, et ODC (*Orthogonal Direction Change*) est le nombre de changements de direction du pointeur par rapport à un axe orthogonal à l'axe optimal de la trajectoire et centré au point de départ de la trajectoire.

Shivani et al., [179] ajoutent à cette liste l'angle moyen de la trajectoire et la vitesse moyenne du mouvement.

Pour faire de la prédiction de mouvements dans un environnement virtuel, Chan et al., [37] ont décomposé le mouvement en une séquence de “pulses” négatifs et positifs le long des axes horizontaux et verticaux. Ils ont observé que la fonction de l'accélération a par rapport à la vitesse v pouvait être approximée par une fonction elliptique :

$$v^2 h_e^2 - 2v l_e h_e^2 + l_e^2 a^2 = 0 \quad (6.6)$$

avec l_e et h_e la largeur et la hauteur de l'ellipse. Cette modélisation caractérise une décomposition des pulses selon deux périodes “symétriques” : une période d'accélération et une période de décélération de même durée. Cela se vérifie pour une tâche de pointage d'une cible mais nous avons expérimentalement vérifié que pour notre tâche d'exploration ce modèle n'est plus valide.

D'une façon générale, les mesures ci-dessus sont pertinentes pour l'évaluation de l'efficacité d'un algorithme d'aide à l'interaction (e.g. [80]) ou d'un périphérique dans le cadre de tâches de déplacement pour lesquelles on connaît *a priori* les trajectoires optimales. En revanche, dans le contexte de fouille visuelle interactive, elles doivent être adaptées.

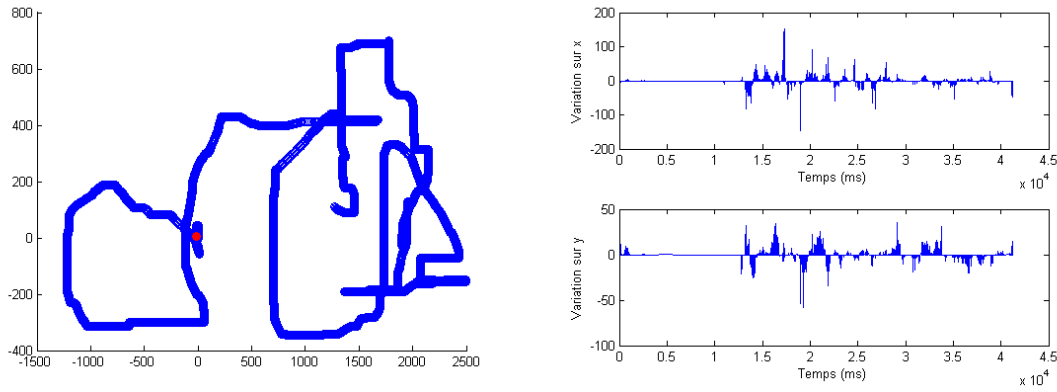


FIGURE 6.2 – Illustration des données de mouvements du pointeur de la souris. A gauche : un exemple de trace d’un mouvement de souris depuis le point d’origine (0,0). A droite : Variations des coordonnées x (en haut) et y (en bas) correspondant à ce tracé en fonction du temps.

6.3 Indicateurs de mouvements pour la détection de communautés

Les paramètres logiciels et matériels de la souris sont restés constants pendant toute la durée de l’expérience. Pour chaque cas, nous avons enregistré la séquence des positions du pointeur de la souris. Les rotations appliquées à un tracé de graphe visualisé se déduisent de sa position : un mouvement horizontal de 10 pixels correspond à une rotation de 1 degré autour du tracé du point de vue de l’utilisateur.

Pour caractériser le mouvement du pointeur de la souris nous avons adapté différents indicateurs décrits dans la section 6.2. Au début de la visualisation d’un tracé, le pointeur est situé à l’origine du système de coordonnées de l’espace d’affichage que nous notons $(0, 0)$. Lorsque le participant bouge la souris à un instant t , le système enregistre t en ms, le déplacement vertical x en pixels et le déplacement horizontal y en pixels. Nous notons $i = 0..n$ l’index des n enregistrements effectués pour un tracé et (x_i, y_i) les positions du pointeur pour le i ème enregistrement. Une fois le mouvement appliqué à la restitution visuelle et enregistré, la position du pointeur de la souris est automatiquement remise à zéro pour autoriser les mouvements infinis dans une même direction. Sans cette précaution, le pointeur pourrait être “bloqué” sur un bord de l’écran. Ces données sont illustrées sur la figure 6.2.

6.3.0.1 Indicateurs de mouvements

Trois indicateurs visent à décrire les mouvements du pointeur :

- D_{tot} : la distance totale (en pixels) parcourue par le pointeur. Elle représente la quantité de mouvement totale. Elle est égale à la somme des distances

euclidiennes parcourues :

$$D_{tot} = \sum_{i=0}^n \sqrt{x_i^2 + y_i^2} \quad (6.7)$$

Elle est définie sur $[0, +\infty]$, et $D_{tot} \rightarrow 0$ si il y a très peu d'interactions et $D_{tot} \rightarrow +\infty$ si au contraire il y en a beaucoup.

- RA : le “ratio d’alternance”. C’est une combinaison des indicateurs MDC (*Movement Direction Change*) et ODC (*Orthogonal Direction Change*) définis section 6.2. Il mesure les changements de direction du mouvement selon l’axe horizontal et l’axe vertical de l’écran. Il est défini par le ratio entre le nombre de changements de direction du pointeur et le nombre de changements de position de ce dernier. Un changement de direction est ici évalué par une fonction de l’angle formé par la droite passant par la position du pointeur et le centre de l’écran et la droite horizontale passant par le centre de l’écran.

$$RA = \frac{\sum_{i=1}^n ang_i}{n - 1} \quad (6.8)$$

avec

$$ang_{i>1} = \begin{cases} 1 & \text{si } \text{sign}(\cos(x_i, y_i)) \neq \text{sign}(\cos(x_{i-1}, y_{i-1})) \\ 1 & \text{si } \text{sign}(\sin(x_i, y_i)) \neq \text{sign}(\sin(x_{i-1}, y_{i-1})) \\ 0 & \text{sinon} \end{cases}$$

Il est défini sur $[0, 1]$, et $RA \rightarrow 1$ si le participant alterne des petits mouvements dans des directions différentes et $RA \rightarrow 0$ si au contraire il effectue de longs mouvements dans une même direction.

- \overline{Courb} : la courbure locale. Elle permet d’évaluer la tendance du mouvement entre une trajectoire linéaire et une trajectoire circulaire, et est une moyenne des courbures définies section 6.2 pour chaque enregistrement.

$$\overline{Courb} = \frac{\sum_{i=2}^n Courb_i}{n} \quad (6.9)$$

avec

$$Courb_i = \frac{|x'_i y''_i - x''_i y'_i|}{(x_i'^2 y_i'^2)^{\frac{3}{2}}}$$

Elle est définie sur $[0, +\infty]$, et $\overline{Courb} \rightarrow 0$ si les interactions correspondent à un mouvement rectiligne et $\overline{Courb} \rightarrow +\infty$ si au contraire les mouvements de la souris forment des courbes très marquées.

6.3.0.2 Indicateurs de la dynamique du mouvement

La modélisation des pauses a été une étape assez délicate car il fallait s'abstraire des “petits” mouvements résiduels éventuels non significatifs. Nous avons procédé au codage des données suivants. Considérons la séquence temporelle M avec l'index $j = 0..t_n$ où t_n correspondant au temps (en ms) auquel le dernier enregistrement a été effectué. Si un mouvement a eu lieu à l'instant j alors $M_j = (x_i, y_i)$ tel que $j = t_i$, sinon $M_j = (0, 0)$. Par exemple, considérons 3 enregistrements qui ont lieu respectivement à $1sec$ où le pointeur est en position $(5, 5)$, $2sec$ en position $(3, 3)$ et $3sec$ en position $(4, 4)$. Nous obtenons $i = 0 \dots 2$ et $j = 0 \dots 3000$ et $M_0 = (0, 0), M_1 = (0, 0), \dots, M_{1000} = (5, 5), M_{1001} = (0, 0), \dots, M_{1999} = (0, 0), M_{2000} = (3, 3), \dots, M_{3000} = (4, 4)$. Cet encodage des données nous permet de réduire le bruit qu'elles contiennent (par exemple des mouvements résiduels de la souris) en calculant le produit de convolution de M avec une gaussienne de taille m :

$$Conv_j = \sum_{g=1}^m G_g M_{j-g+1}$$

avec $G \approx \mathcal{N}(0, 1)$

L'instant j est considéré comme une pause si $p_j = 1$ avec :

$$p_j = \begin{cases} 1 & \text{si } \sum_j^{j+\theta} |Conv_j| \leq \delta \\ 0 & \text{sinon} \end{cases} \quad (6.10)$$

Nos expérimentations nous ont conduit à choisir $m = 50$ et $\delta = 2$ et $\theta = 100$, ce qui est cohérent avec ce que [138] considère comme la durée minimale d'un mouvement conscient. Nous avons expérimentalement testé la robustesse des résultats au voisinage de ces valeurs.

Quatre indicateurs permettent de caractériser l'influence des pauses dans la dynamique :

- P_{tot} : le nombre de pauses effectué par un participant pour l'exploration d'un tracé. Il est défini par le nombre de sous-séquences distinctes de $p_j = 1$. Il est défini sur $[0, +\infty]$, et $P_{tot} \rightarrow 0$ si il y a très peu de pauses dans l'interaction et $P_{tot} \rightarrow +\infty$ si au contraire le nombre de pauses est très important.
- RP : le ratio temps pause / temps total. Il est défini par la proportion du temps passé en pause par rapport au temps total d'interaction.

$$RP = \frac{\sum_{j=0}^{t_n} p_j}{t_n} \quad (6.11)$$

Il est défini sur $[0, 1]$, et $RP \rightarrow 0$ si les pauses représentent une infime partie du temps d'interaction et $RP \rightarrow 1$ si au contraire la majorité du temps

d'interaction est passé en pause.

- \overline{P} : la durée moyenne d'une pause (en secondes) qui est définie par :

$$\overline{P} = \frac{\sum_{j=0}^{t_n} p_j}{P_{tot}} \quad (6.12)$$

Elle est définie sur $[0, +\infty]$, et $\overline{P} \rightarrow 0$ si les pauses sont très courtes et $\overline{P} \rightarrow +\infty$ si au contraire les pauses sont très longues.

- $\overline{\Delta P}$: la période moyenne entre deux pauses (en secondes). Elle représente le temps moyen qui se déroule entre deux pauses consécutives et est définie comme suit.

$$\overline{\Delta P} = \frac{\sum_{j=0}^{t_n} |p_j - 1|}{P_{tot} - 1} \quad (6.13)$$

Elle est définie sur $[0, +\infty]$, et $\overline{\Delta P} \rightarrow 0$ si les pauses sont très fréquentes et $\overline{\Delta P} \rightarrow +\infty$ si au contraire il faut attendre longtemps entre deux pauses successives.

6.4 Analyse des Mouvements

Cette section rapporte l'analyse des mouvements observés dans notre expérimentation en se basant sur les indicateurs ci-dessus.

6.4.1 Approche exploratoire : classification non supervisée

Dans une démarche exploratoire, sans modèle préalable des comportements, nous avons cherché à savoir s'il était possible de discriminer les deux conditions expérimentales à partir des indicateurs de mouvements. Chaque tracé visualisé dans chaque condition est ici décrit par l'ensemble des indicateurs $\{D_{tot}, RA, P_{tot}, RP, \overline{D}, \overline{\Delta P}, \overline{Courb}\}$ préalablement normalisés et est associé à une condition expérimentale $Cond = \{-1, 1\}$ (resp. 3D monoscopique et 3D stéréoscopique). On a ainsi en entrée de l'analyse $207 \text{ tracés} \times 2 \text{ conditions} = 414$ enregistrements.

La catégorisation est effectuée par un algorithme de machine à vecteurs de support (SVM) classique avec en entrée une base d'apprentissage de 70% des enregistrements. Dix tests ont été effectués avec plusieurs noyaux d'apprentissage. Et les meilleurs résultats ont été obtenus avec un taux moyen de catégorisation correct de 70% ($\sigma = 2.5$) avec un noyau quadratique.

Ces résultats indiquent qu'il semble être possible de déterminer *a posteriori* si un graphe a été visualisé en 3D mono ou en 3D stéréo en utilisant uniquement nos indicateurs. Cela suggère qu'il existe vraisemblablement des différences dans l'interaction entre les deux conditions mais ne nous permet pas de les caractériser.

Nous avons donc procédé à une analyse plus fine qui nous a permis d'identifier l'impact discriminant des indicateurs.

6.4.2 Analyse statistique des indicateurs

Dans cette section, nous présentons une analyse statistique plus approfondie des différents indicateurs. Pour chacun d'entre eux, nous dressons un tableau récapitulatif complété par le résultat d'un test statistique visant à déterminer si la différence entre les deux conditions expérimentales est significative. Pour chacun de ces tests, nous avons préalablement vérifié si les échantillons suivaient une loi normale en utilisant le test de Kolmogorov-Smirnov [128]. Aucune des distributions des indicateurs ne remplissant cette condition, nous avons utilisé un test de rangs de Wilcoxon [205]. Ce test non paramétrique est adapté aux échantillons appariés que nous avons construits dans notre protocole (à une observation en mono est associée une observation en stéréo avec le même tracé et le même utilisateur).

Distance totale D_{tot} (éq. 6.7, tab. 6.1) La distance totale parcourue par le pointeur semble plus grande en mono. Cependant, cette différence observée n'est pas statistiquement significative ($Wilcoxon(206) = 44893$, $z_{value} = 1.5937$, $p_{value} = 0.1$).

Condition	min	max	avg	std
3D Stéréo	682	45200	7398	6371
3D Mono	564	38143	7651	5719

TABLE 6.1: Statistiques pour l'indicateur D_{tot} représentant la distance totale parcourue par la souris. Les valeurs sont exprimées en pixels.

Ratio d'alternance RA (éq. 6.8, tab. 6.2) Il représente la fréquence à laquelle le pointeur de la souris change de direction. La stéréo entraîne une augmentation de 22% des changements de direction par rapport à la condition mono. Cette différence est statistiquement significative ($Wilcoxon(206) = 35157$, $z_{value} = -6.4035$, $p_{value} < 0.001$).

Condition	min	max	avg	std
3D Stéréo	0.14	0.79	0.38	0.11
3D Mono	0.07	0.62	0.31	0.10

TABLE 6.2: Statistiques pour l'indicateur RA représentant le ratio entre le nombre de changements de direction de la souris par rapport au nombre de changements de position.

Courbure locale \overline{Courb} (éq. 6.9, tab. 6.3) Elle permet de déterminer si les mouvements de souris suivent une trajectoire courbée ou une trajectoire plus rectiligne. La stéréo a tendance à produire des trajectoires plus courbées que la mono ; mais cette différence n'est pas significative ($Wilcoxon(206) = 42679$, $z_{value} = -0.2243$, $p_{value} = 0.82$).

Condition	min	max	avg	std
3D Stéréo	0.0008	3.65	0.14	0.35
3D Mono	0.0001	2.29	0.14	0.29

TABLE 6.3: Statistiques pour l'indicateur \overline{Courb} représentant la courbure locale moyenne du mouvement.

Nombre de pauses P_{tot} (éq. 6.10, tab. 6.4) Il représente le nombre d'interruptions des interactions détectées lors de l'interaction avec les tracés. En moyenne, ces pauses sont plus fréquentes en stéréo qu'en mono, mais cette observation n'est pas statistiquement significative. ($Wilcoxon(206) = 41389$, $z_{value} = -1.2849$, $p_{value} = 0.19$).

Condition	min	max	avg	std
3D Stéréo	2	176	20.35	22.27
3D Mono	2	85	16.52	13.61

TABLE 6.4: Statistiques pour l'indicateur P_{tot} représentant le nombre de pauses identifiées dans les interactions.

Ratio entre le temps passé en pause et le temps total RP (éq. 6.11, tab. 6.5) Il permet, à partir des pauses détectées, de calculer la proportion du temps passée par les utilisateurs à observer le graphe par rapport au temps passé à interagir activement. Le temps passé en pause, est plus important en stéréo qu'en mono, et cette différence est significative ($Wilcoxon(206) = 39193$, $z_{value} = -3.0879$, $p_{value} = 0.002$).

Condition	min	max	avg	std
3D Stéréo	0.14	0.89	0.53	0.15
3D Mono	0.17	0.86	0.49	0.14

TABLE 6.5: Statistiques pour l'indicateur RP représentant le rapport entre le temps passé en pause et le temps total.

Durée d’une pause \bar{P} (éq. 6.12, tab. 6.6) Les pauses semblent être légèrement plus longues en stéréo, mais cette différence n’est pas statistiquement significative ($Wilcoxon(206) = 41909$, $z_{value} = -0.8572$, $p_{value} = 0.3$).

Condition	min	max	avg	std
3D Stéréo	0.13	2.5	0.69	0.43
3D Mono	0.19	1.7	0.62	0.30

TABLE 6.6: Statistiques pour l’indicateur \bar{P} représentant la durée moyenne d’une pause, en secondes.

Période entre deux pauses $\overline{\Delta P}$ (éq. 6.13, tab 6.7) Elle représente le temps moyen écoulé entre deux pauses successives. Cette période est plus courte en stéréo de manière significative ($Wilcoxon(206) = 46182$, $z_{value} = 2.6526$, $p_{value} = 0.008$).

Condition	min	max	avg	std
3D Stéréo	0.1	4.33	0.62	0.50
3D Mono	0.18	3.51	0.66	0.44

TABLE 6.7: Statistiques pour l’indicateur $\overline{\Delta P}$ représentant la période moyenne entre deux pauses, en secondes.

6.4.3 Discussion

L’analyse exploratoire des interactions avec notre système de visualisation de graphes confirme l’hypothèse selon laquelle les utilisateurs ne manipulent pas les tracés de graphes de la même manière en fonction de la méthode de restitution.

Plus précisément, l’analyse statistique des indicateurs nous permet de découvrir certaines tendances principales. D’après nos résultats, un utilisateur sous la condition stéréoscopique marque plus fréquemment des pauses dans la manipulation du tracé et change plus souvent la direction de la trajectoire du mouvement de souris. Réciproquement, un utilisateur sous la condition monoscopique effectue moins de pauses dans la manipulation et a tendance à effectuer des mouvements de souris associés à une trajectoire plus “continue”. Ces observations confirment les hypothèses 1, 2 et 3 formulées dans la section 6.1. En revanche, l’hypothèse 4 est réfutée, la distance totale parcourue entre les deux conditions n’est pas significativement différente.

Cette étude confirme qu’il existe vraisemblablement des différences de comportements dépendants de la condition de restitution visuelle. Cependant, les indica-

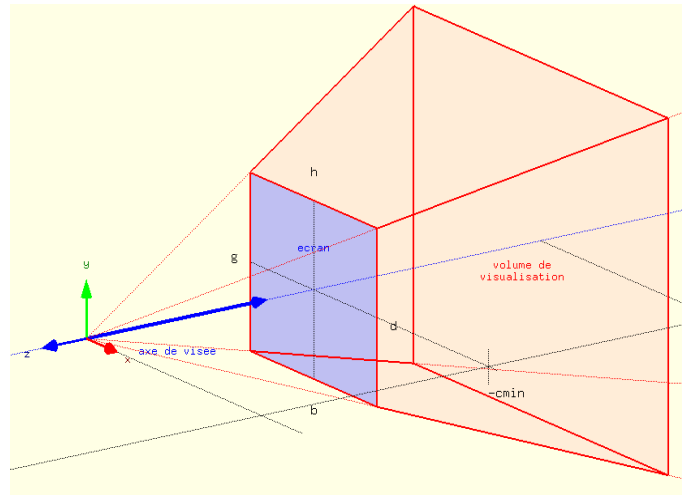


FIGURE 6.3 – Illustration d'un point de vue sur le graphe. Par ses interactions, l'utilisateur contrôle la position de son point de vue autour du graphe dans l'espace graphique. Ce dernier se trouve au centre du volume de visualisation.

teurs de mouvements du périphérique ne permettent pas de comprendre l'impact sur la tâche. Nous avons donc cherché à caractériser les zones qui étaient perçues sur les tracés au cours de l'exploration.

6.5 Analyse des points de vue

Un point de vue représente la caméra virtuelle que contrôle l'utilisateur avec ses interactions (figure 6.3). Il est associé à deux attributs pertinents pour notre analyse : une position dans l'espace graphique et un vecteur directionnel qui a pour origine cette position et qui est orienté vers le centre du tracé. À partir de ces enregistrements, nous pouvons retracer *a posteriori* la séquence des zones du tracé vues par l'utilisateur au cours de son exploration.

6.5.1 Carte de chaleur 3D

Pour chaque tracé sous chaque condition, nous représentons les temps passés sur les zones perçues par une carte de chaleur sur l'enveloppe sphérique du tracé. Le calcul de cette carte a nécessité une triangulation préalable de l'enveloppe du tracé. Plus précisément, nous avons calculé la triangulation de la sphère englobante de rayon minimum de chaque tracé.

Pour se faire, nous avons tout d'abord produit une approximation polygonale de cette sphère par une géode (voir figure 6.4) de même rayon que la sphère englobante. Chaque triangulation comporte 8000 triangles de tailles égales ; ce qui est un bon compromis entre le temps de calcul de la triangulation et la précision requise. À chacun de ces triangles nous associons une valeur tr_k initialement nulle avec $k =$

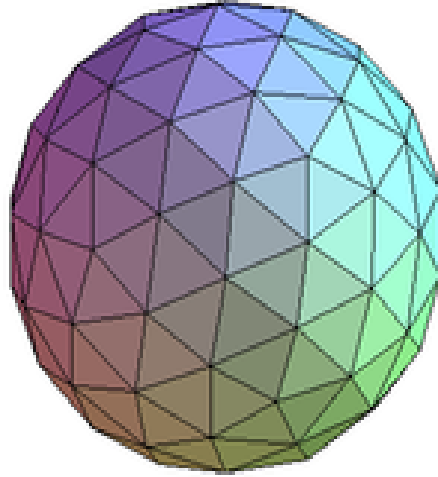


FIGURE 6.4 – Illustration d’une géode. Cette structure géométrique nous permet d’obtenir une approximation polygonale de la sphère englobante de chaque tracé afin de pouvoir construire une carte de chaleur sur une surface discrétisée.

1..8000.

Puis, pour chaque point de vue pris d’un participant sur un tracé, nous avons identifié l’ensemble des triangles “visibles” qui appartiennent au cône défini par la position du point de vue et les tangentes à la sphère passant par ce point. Nous augmentons alors la valeur tr_k des triangles visibles par la durée (en ms) durant laquelle l’utilisateur est resté sur ce point de vue. Ces durées peuvent ensuite être représentées sur une carte de chaleur (Fig. 6.5) : les zones rouges (resp. bleues) correspondent aux zones du tracé qui ont été plus souvent (resp. moins souvent) dans le champ de vision du participant.

6.5.2 Résultats

Pour comparer les comportements utilisateurs à une échelle macroscopique, nous avons mesuré l’entropie de Shannon [177] associée à chaque carte de chaleur :

$$H = - \sum_{k=1}^q p(tr_k) \log p(tr_k) \quad (6.14)$$

avec q le nombre de valeurs distinctes parmi les différentes instances tr_k et $p(tr_k)$ les fréquences observées qui leur sont associées. Les valeurs élevées de l’entropie correspondent à une distribution uniforme de ces valeurs. L’entropie moyenne sur les 207 tracés retenus est respectivement de $H_{mono} = 5.12$ pour la 3D monoscopique et de $H_{stereo} = 4.67$ pour la 3D stéréoscopique. Un test de rangs de Wilcoxon confirme que la 3D monoscopique est associée à des valeurs plus élevées

de l'entropie de manière significative ($Wilcoxon(206) = 14939$, $z_{value} = 8.66$, $p_{value} < 0.0001$).

Ces résultats confirment l'existence d'une différence comportementale entre les restitutions monoscopique et stéréoscopique pour une tâche de détection de communautés. En 3D stéréoscopique, l'exploration du tracé semble concentrée sur des régions plus restreintes avec une zone importante peu explorée. En 3D monoscopique, les participants explorent l'intégralité du tracé affiché à l'exception de petites zones "d'angles morts". La figure 6.5 illustre ces différences. L'histogramme de la distribution des valeurs de temps pour la 3D monoscopique suggère une tendance vers une distribution uniforme ; ce qui correspond à la coloration quasi-homogène de la carte de chaleur (à l'exception d'une petite zone ignorée en bleu foncé). *A contrario*, on distingue sur l'histogramme associé à la 3D stéréoscopique un nombre plus grand de petites valeurs ainsi qu'un pic de valeurs élevées. Sur la carte de chaleur, ces valeurs sont associées à une large zone bleue et à une petite zone rouge sur laquelle le participant a concentré son attention. L'analyse de ces comportements complète les observations décrites dans la section 6.4. La présence de pauses plus grandes en 3D stéréoscopique coïncide avec l'existence de zones visuelles sur lesquelles un participant concentre son attention. Les changements de direction, plus fréquents en 3D stéréoscopique et associés à des mouvements produisant des points de vue proches, sont cohérents avec cette analyse.

Discussion

Dans ce chapitre, les résultats de deux études ont mis en évidence l'existence des différences de comportement dans les interactions avec les tracés de graphes dans une tâche de détection de communautés. Nous avons tout d'abord identifié certaines tendances au niveau des mouvements de souris qui nous ont permis de découvrir des stratégies d'exploration bien distinctes de l'exploration des graphes. En stéréoscopie, les participants ont plutôt tendance à se concentrer sur des zones spécifiques du graphe, tandis qu'en monoscopie ils ont tendance à explorer plus globalement l'espace. Il est intéressant de mettre ces résultats en parallèle avec les résultats obtenus dans le chapitre précédent. La stéréoscopie semblant donner de meilleures performances pour la détection de communautés, ces différences d'interaction suggèrent que la stratégie mise en œuvre dans la condition stéréoscopique est efficace. En d'autres termes, les participants sont capables, grâce à la stéréoscopie, d'identifier les différentes communautés d'un graphe (sujettes, ou non, à l'occlusion) à partir d'un nombre plus restreint de points de vues. Cette propriété, si elle était confirmée, serait cohérente avec l'une des propriétés de la stéréoscopie

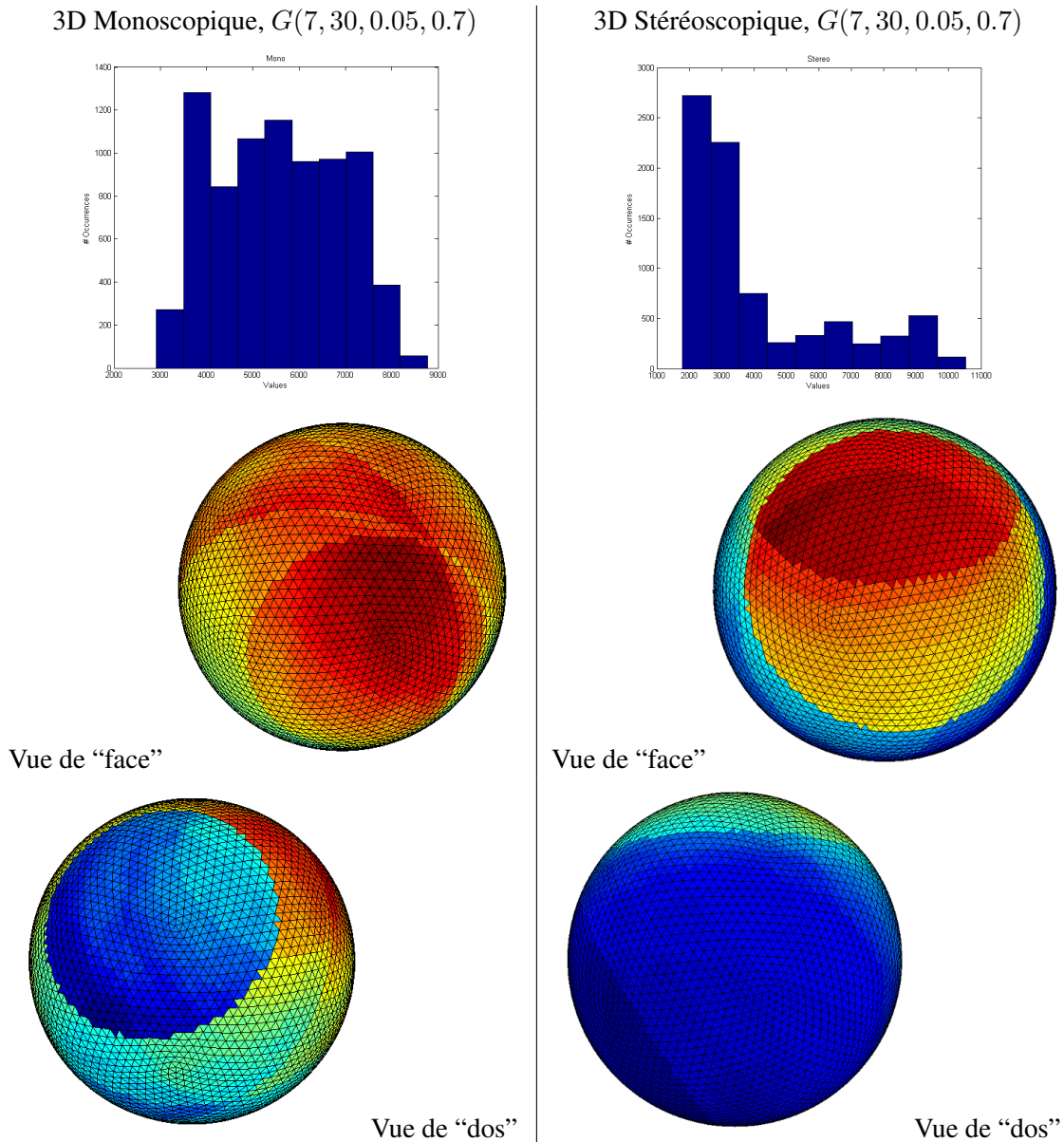


FIGURE 6.5 – Exemple de deux vues complémentaires de carte de chaleur pour les conditions monoscopique (gauche) et stéréoscopique (droite), ainsi que les distributions associées (abscisses : valeurs de temps, ordonnée : effectif). La distribution observée en monoscopie est plus uniforme qu’en stéréoscopie. En stéréoscopie le participant concentre son attention sur une zone plus restreinte. Une zone particulièrement étendue semble avoir pratiquement été ignorée.

énoncée dans le chapitre 2 : la stéréoscopie facilite la visualisation d'objets à travers un désordre visuel important.

Vers de nouveaux modes interactifs

GUIs are showing clear signs of limitations as they face the ever-increasing reach of the digital revolution.

—Michel Beaudouin-Lafon

Sommaire

Introduction	108
7.1 Interactions dans les logiciels de fouille	109
7.2 Déclenchement des interactions dans <i>TempoSpring</i>	110
7.3 Interface mains libres dans <i>TempoSpring</i>	112
7.3.1 Principes généraux	112
7.3.2 Automate	113
7.3.3 Interactions multimodales	116
Conclusion	117

Introduction

Les interactions entre un système de visualisation et ses utilisateurs sont au cœur du processus de fouille visuelle. Elles permettent de modifier la restitution des données et de déclencher des mécanismes analytiques pour diriger l'extraction de connaissances. Dans la grande majorité des cas, ces interactions sont déclenchées en utilisant conjointement une souris (ou un clavier) et une interface graphique. Ce paradigme d'interaction a été inventé dans les années 70 par les ingénieurs de Xerox puis démocratisé par Apple et Microsoft dans les années 80. Il est aujourd'hui utilisé dans la grande majorité des systèmes informatiques à travers le monde, mais comme le souligne Beaudouin-Lafon [18], il commence à atteindre ses limites. Celles-ci peuvent être catégorisées selon trois points :

- Difficulté du passage à l'échelle : les applications informatiques permettent de manipuler des volumes de données toujours plus importants, mais les interactions restent identiques. En visualisation de graphes par exemple, il peut être approprié de sélectionner manuellement les nœuds à filtrer dans une liste si leur nombre n'excède pas quelques dizaines. Mais ce procédé est invisable si le graphe comporte des millions de nœuds.
- Difficulté du portage vers d'autres plateformes : les interfaces graphiques ont été optimisées pour un environnement spécifique comportant une souris, un clavier et un écran d'ordinateur. Mais l'informatique d'aujourd'hui ressemble d'avantage à la vision de Weiser [204] de "l'informatique ubiquitaire" avec des dispositifs interconnectés aux spécificités différentes (téléphones intelligents, tablettes, murs et tables interactifs). Dans ce contexte, il est très rare de pouvoir effectuer la même action avec les mêmes interactions sur des dispositifs différents.
- Interopérabilité limitée et manque de flexibilité : les interactions d'un système sont la plupart du temps strictement spécifiées en amont de son développement et intégrées à son interface, ce qui limite les possibilités d'utilisation innovantes de la part des utilisateurs. En sus, elles sont la plupart du temps cantonnées au cadre de l'application. Il est par exemple impossible d'utiliser une palette de couleurs d'un logiciel de retouche d'images pour modifier l'aspect des nœuds dans une application de visualisation de graphes ou pour changer la couleur d'un menu.

Pour palier à ces limites, Beaudouin-Lafon propose un nouveau paradigme d'"interaction instrumentale" pour lequel une interaction n'est pas définie dans le contexte spécifique d'une interface, mais comme un outil que l'utilisateur manipule directement. Par exemple selon ce paradigme, un utilisateur pourrait utiliser un outil "loupe" pour agrandir une image ou un texte sans avoir à chercher les boutons correspondant dans une visionneuse d'image et dans un logiciel de traitement de texte.

Dans ce chapitre, nous nous plaçons dans un cadre de plus en plus commun [163] de visualisation dans un environnement immersif sur grand écran et nous nous inspirons des préceptes de l'interaction instrumentale et des trois points cités ci-dessus pour proposer une méthode d'interaction novatrice avec *TempoSpring*. Plus précisément, nous présentons une bibliothèque permettant, via le capteur *Kinect* de Microsoft, d'interagir avec la restitution sans manipuler de périphériques. Ce système permet à l'utilisateur d'appliquer une rotation, de zoomer, de filtrer ou d'agréger des nœuds à partir de gestes très simples des mains ou des bras. Ce type d'interaction est souvent qualifié de “mains libres” puisque l'utilisateur ne doit s'encombrer d'aucun périphérique exogène pour les mener à bien.

Nous identifions tout d'abord certains travaux visant à adapter les interactions d'un système de fouille visuelle à un environnement immersif. Ensuite, nous rappelons les interactions principales proposées par *TempoSpring* et la façon dont elles sont déclenchées. Puis nous décrivons le fonctionnement de notre bibliothèque d'interaction et son intégration dans *TempoSpring*.

7.1 Interactions dans les logiciels de fouille visuelle immersive

Comme nous l'avons déjà mentionné, le problème de la complexité visuelle des restitutions est récurrent en fouille visuelle de graphes. Le plus souvent, des interactions sont proposées à l'utilisateur en plus d'une agrégation hiérarchique permettant de réduire la complexité visuelle. Ces interactions permettent à l'utilisateur de manipuler la représentation visuelle du graphe, de la filtrer, et d'obtenir des informations détaillées sur certaines parties spécifiques. La grande majorité des systèmes reposent sur le paradigme d'interaction classique : WIMP (*Windows Icons Menus Pointer*). Mais cette approche est peu adaptée dans des environnements immersifs où l'utilisation d'un clavier et d'une souris n'est pas idéal pour interagir avec un espace 3D. Certaines approches alternatives ont été proposées dans la littérature.

Par exemple, Osawa et al. [151] ont proposé en l'an 2000 une interface basée sur des gestes reconnus par le biais de deux capteurs magnétiques pour interagir avec un graphe. Mais il y a plus de 10 ans, le matériel d'acquisition existant était trop limité pour permettre de reconnaître un large éventail de mouvements. Plus récemment, Halpin et al. [92] ont implémenté une baguette virtuelle, contrôlée par un appareil de type *Wand* ressemblant à une télécommande, dans leur outil de visualisation de réseaux sociaux. Cette baguette virtuelle permet de sélectionner un nœud dans la restitution et l'extruder vers l'utilisateur. Comme nous l'avons mentionné dans le chapitre 2, le graphe est initialement affiché en 2D, mais le nœud sélectionné et ses

voisins sont affichés en 3D stéréoscopique ce qui donne l'impression à l'utilisateur qu'ils "sortent de l'écran". Rappelons que les auteurs indiquent que ce paradigme d'interaction, couplé à l'usage de la stéréoscopie favorise l'analyse fine des graphes.

La navigation et l'interactivité dans un environnement virtuel affiché sur un grand écran a été largement étudié par la communauté IHM. Deux grandes familles d'approches émergent de ces travaux : (i) celles qui dépendent de périphériques classiques (souris, clavier) ou plus récents (Wiimote, souris 3D, écran tactile annexe etc..) et (ii) celles qui ne dépendent que du suivi du corps de l'utilisateur [144]. Par exemple, Xia et al. [208] présentent un système de simulation chirurgical contrôlé par une souris 3D tandis que Zigelbaum et al. [210] présentent un système multimédia intégralement contrôlé par une interface "mains libres". La question de l'évaluation de ces interfaces reste largement ouverte et sort du cadre de cette thèse.

7.2 Déclenchement des interactions dans *TempoSpring*

Dans cette partie, nous rappelons les interactions principales fournies par *TempoSpring* (chapitre 4), et pour chacune d'entre elles, nous indiquons la manière dont elles sont déclenchées lorsque la bibliothèque d'interaction immersive n'est pas utilisée.

1. *rotation* pour découvrir les parties non visibles du graphe l'utilisateur peut appliquer une rotation centrée sur son barycentre. Pour déclencher cette interaction, l'utilisateur doit maintenir le bouton droit de la souris enfoncé tout en déplaçant celle-ci.
2. *zoom* pour visualiser l'intégralité du graphe avec un grand angle de vue ou pour se concentrer sur des caractéristiques locales, l'utilisateur peut zoomer/dezoomer. Cette interaction est déclenchée en utilisant la molette de la souris.
3. *filtrage* pour simplifier le graphe et surligner certaines propriétés structurelles, différents filtres sont disponibles : un filtrage sur les degrés permettant de cacher les nœuds et liens isolés afin de ne visualiser que les nœuds structurant du graphe, et un filtrage surlignant les nœuds et liens sujets aux forces les plus importantes. Ces filtrages sont déclenchés via des raccourcis clavier. Certains paramètres peuvent être modifiés en utilisant des curseurs affichés dans la restitution. Pour les utiliser, il suffit de cliquer sur la partie jaune du curseur et, tout en gardant le bouton de souris enfoncé, de la déplacer.
4. *agrégation manuelle* pour former une agrégation hiérarchique l'utilisateur peut sélectionner un ensemble de nœuds pour les rassembler en un seul méta-nœud. Pour déclencher cette interaction, l'utilisateur doit maintenir le bouton

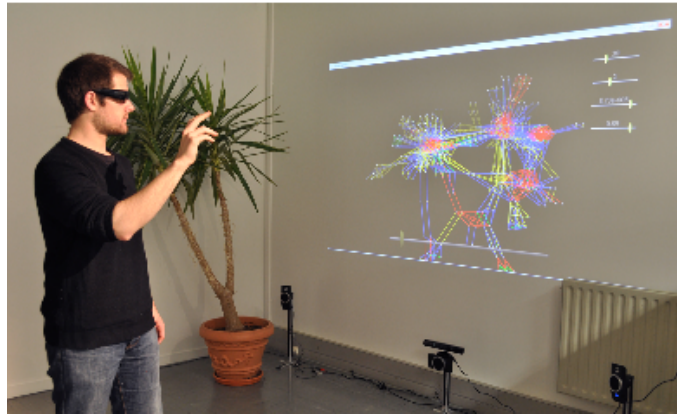


FIGURE 7.1 – Photo d’un utilisateur explorant un réseau social affiché sous forme de graphe en 3D stéréoscopique sur un mur par *TempoSpring*. Dans cette configuration, la restitution représente une surface de $2,30 \times 1,30 \text{ m}^2$, l’utilisateur est debout et l’utilisation d’un clavier et d’une souris est peu adaptée.

shift et le bouton gauche de la souris enfoncés tout en déplaçant cette dernière. Ceci a pour effet de faire apparaître une fenêtre de sélection dans la restitution. L’agrégation est validée en appuyant sur un raccourci clavier tout en relâchant le bouton gauche de la souris.

Le moteur de rendu OpenGL 3D au cœur *TempoSpring* offre la possibilité d’afficher des restitutions en 3D monoscopique, ainsi qu’en 3D stéréoscopique active ou passive. Associé à un projecteur, il permet donc la projection des graphes en 3D stéréoscopique sur une surface très large. Comme nous l’avons mentionné dans le chapitre 2, certaines études montrent que l’utilisation de grands écrans est bénéfique pour la fouille visuelle [146, 163] et que la stéréoscopie semble proposer des améliorations prometteuses pour différentes tâches. D’après notre expérience, combiner une restitution stéréoscopique et un grand écran entraîne un sentiment d’immersion convaincant. Ces observations sont d’ailleurs partiellement confirmées par [105] et [100].

Dans ce contexte, l’utilisateur se tient debout devant un écran/mur sur lequel le graphe est affiché. Dans cette position, les périphériques classiques comme les souris et les claviers ne sont pas adaptés. De plus, dans un souci d’optimisation de l’espace d’affichage, nous avons limité le nombre d’effecteurs dans la fenêtre de visualisation. En contrepartie, nous avons eu recours à de nombreux raccourcis clavier pour déclencher les différentes interactions proposées par *TempoSpring*. Ceci entraîne une surcharge cognitive pour l’utilisateur qui doit retenir tous ces déclencheurs.

Ces deux raisons nous ont amené à proposer un moteur d’interactions mains libres qui permet aux utilisateurs de déclencher des interactions debout (voir la figure 7.1).

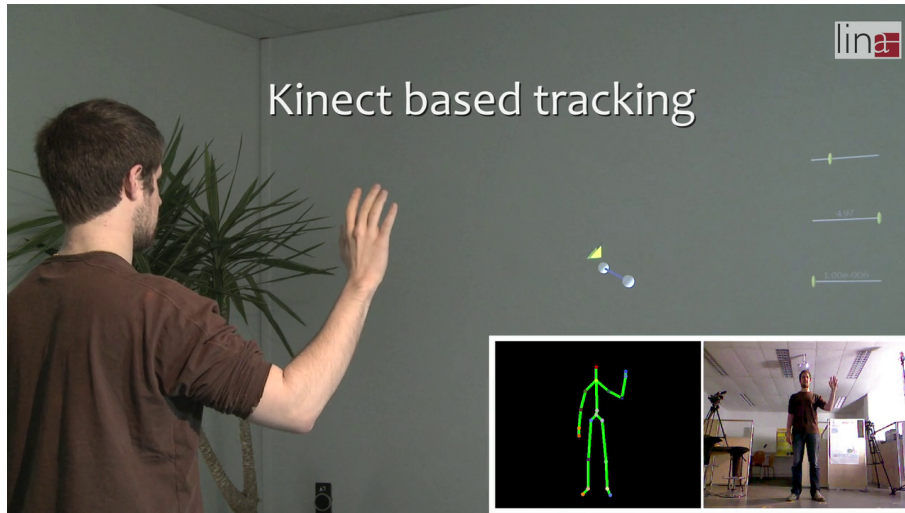


FIGURE 7.2 – Illustration du *Kinect*. Ce capteur est équipé d’une caméra classique ainsi que d’un capteur de profondeur qui permet d’acquérir la carte de profondeur de l’environnement filmé. A partir de ces deux sources de données, le *Kinect* est capable d’extraire un ou des “squelette(s)” représentant le ou les utilisateurs. Il est ensuite possible d’accéder en temps réel aux positions des différents points d’intérêt de ce squelette (les articulations, mains, têtes etc...) pour détecter différents mouvements.

7.3 Interface mains libres dans *TempoSpring*

Stimulés par la croissance de l’industrie vidéo-ludique, de nouvelles technologies de capteurs fiables et abordables ont émergé ces dernières années. Une illustration paradigmatique est le *Kinect* de Microsoft. Ces capteurs ont permis le développement de nouvelles interfaces utilisateurs qui permettent une plus grande liberté de mouvement dans l’interaction que les périphériques habituels (la position assise devant des périphériques d’interaction n’étant plus requise). Afin de tirer parti de ces technologies en fouille visuelle de graphes, nous avons adapté les interactions décrites dans la section 7.2 à des gestes pour le *Kinect*. Le fonctionnement de ce capteur est résumé sur la figure 7.2.

7.3.1 Principes généraux

Les gestes déclencheurs des interactions sont associés aux positions des mains de l’utilisateur : (i) avec la main gauche levée, l’utilisateur peut explorer l’espace en appliquant une rotation au graphe en fonction du mouvement de sa main (figure 7.3) ; (ii) avec la main droite levée, un pointeur apparaît dans la fenêtre de visualisation qui permet d’interagir avec les différents effecteurs (figure 7.4) ; (iii) le geste désormais familier du “*pinch to zoom*” (pincer pour zoomer) est imité pour zoomer et dezoomer en rapprochant ou en éloignant ses deux mains en l’air. Avec les deux mains en l’air, l’utilisateur peut également sélectionner un groupe de nœuds en

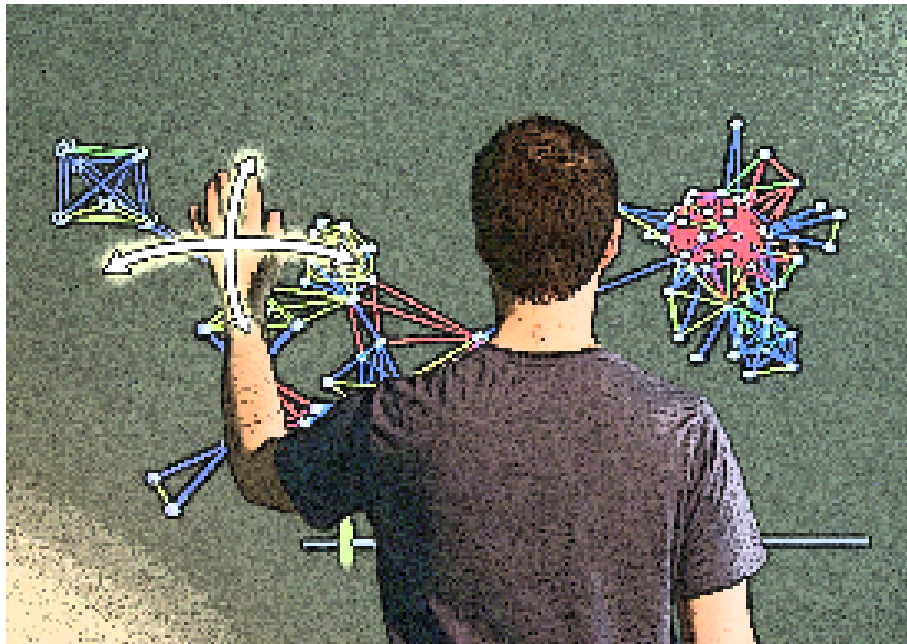


FIGURE 7.3 – Dans l’état d’exploration, le mouvement de la main entraîne une rotation du graphe centrée sur son barycentre.

contrôlant la position des coins haut-gauche et bas-droite d’une fenêtre de sélection superposé au graphe.

Deux gestes additionnels sont reconnus par notre système : un geste de “swipe” (de gauche à droite ou de droite à gauche, voir figure 7.5) et un “clic” qui consiste à lever puis rabaisser sa main très rapidement (figure 7.6). Le “swipe” est associé à l’activation des filtres tandis que le clic permet de sélectionner un nœud survolé par le pointeur virtuel.

Des commandes vocales sont également reconnues. Elles permettent par exemple d’actionner l’agrégation d’un groupe de nœuds sélectionnés (figure 7.7) ou de transmettre un méta-nœud sélectionné à un écran secondaire (figure 7.8).

7.3.2 Automate

Certains des gestes décrits ci-dessus ne sont pas indépendants : avec les deux mains levées, on peut soit déclencher un zoom, soit une sélection de nœuds. Pour éviter ce problème et améliorer *TempoSpring*, nous avons développé le moteur d’interactions en utilisant un automate fini. Ses deux états principaux sont l’état d’exploration (“*browse state*”) et l’état de sélection (“*select state*”) déclenchés en levant respectivement la main gauche et la main droite. Deux états supplémentaires viennent les compléter en levant l’autre main : l’état de zoom (“*zoom state*”) et l’état de sélection multiple (“*multiple selection state*”). Le comportement de cet automate est illustré sur la figure 7.9. Dans les états d’exploration et de sélection, le moteur d’interactions suit activement la vitesse de déplacement de la main qui n’est pas le-

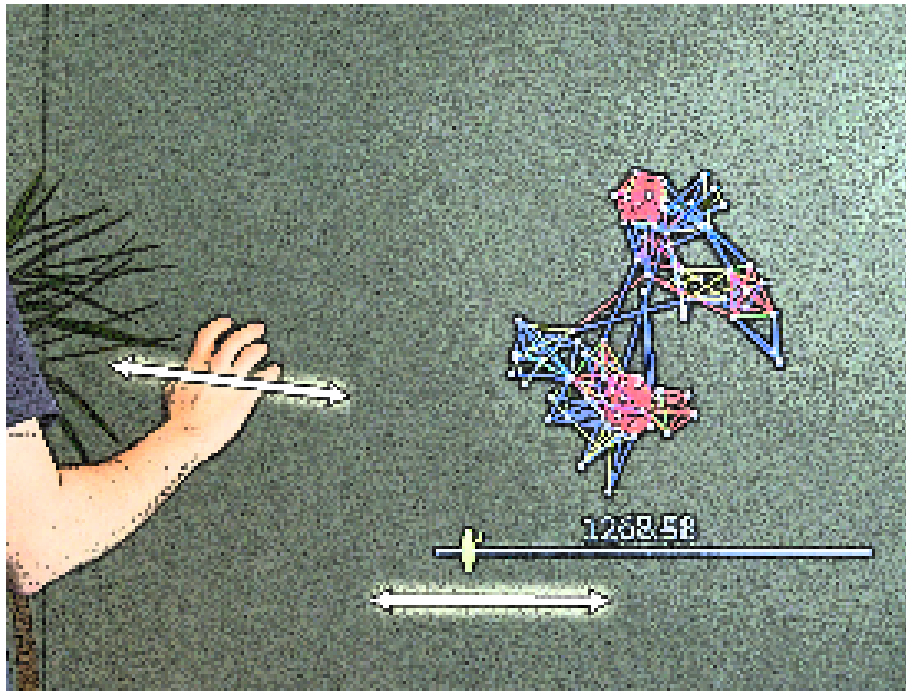


FIGURE 7.4 – Dans l'état de sélection, un pointeur virtuel (un triangle jaune) peut être utilisé pour interagir avec les curseurs. Dans ce cas, il permet d'explorer la ligne temporelle d'un graphe dynamique.

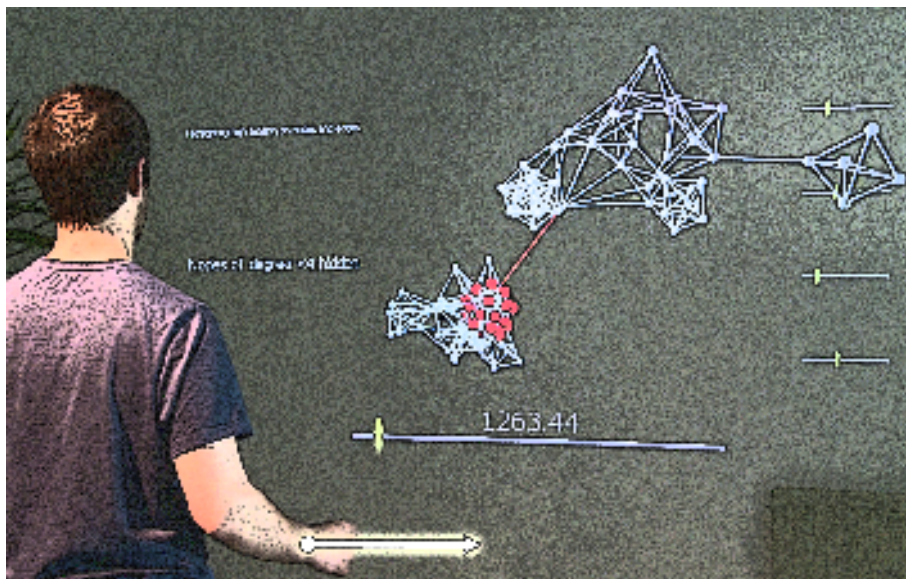


FIGURE 7.5 – Un “swipe” permet à l'utilisateur d'activer ou de désactiver les mécanismes de filtrage. Ici, les nœuds surlignés apparaissent en rouge tandis que les autres apparaissent en gris.

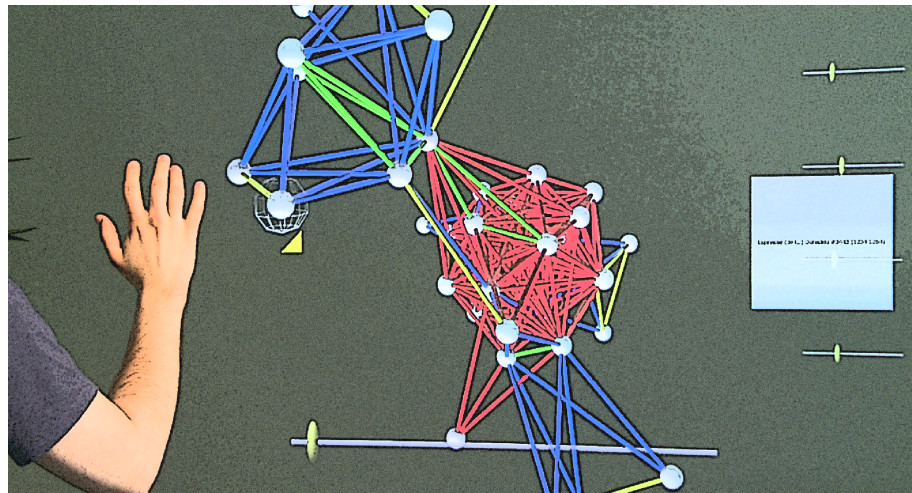


FIGURE 7.6 – Dans l'état de sélection, l'utilisateur peut sélectionner un nœud et effectuer un geste de clic pour afficher le contenu associé à ce nœud (il apparaît dans la fenêtre à la droite de la photo).

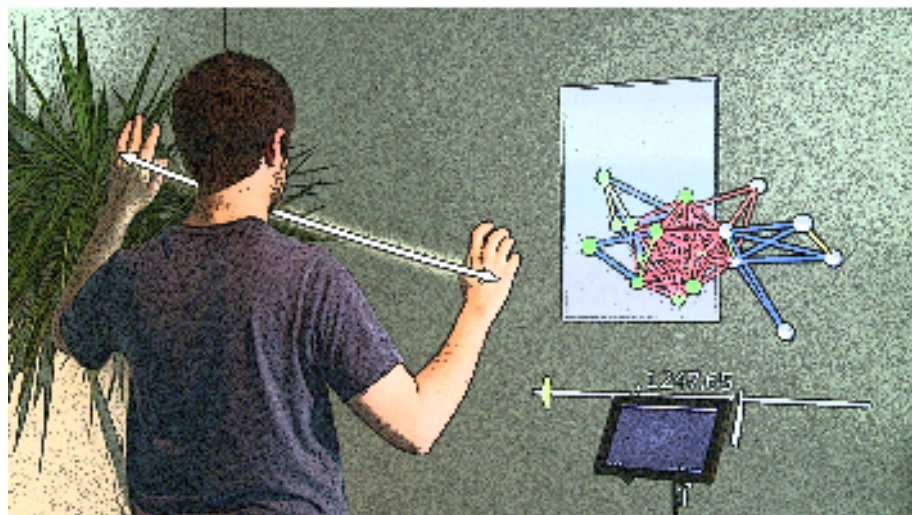


FIGURE 7.7 – Dans l'état de sélection multiple, l'utilisateur peut contrôler une fenêtre superposée au graphe pour sélectionner un ensemble de nœuds. Une fois sélectionnés, ceux-ci peuvent être agrégés en un méta-nœud via une commande vocale.

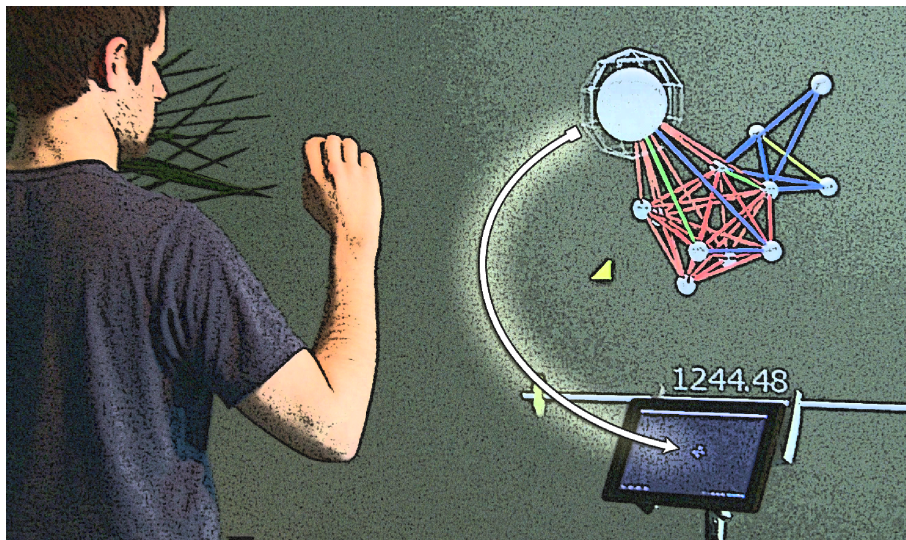


FIGURE 7.8 – Si un méta-nœud est sélectionné, il peut être transmis à un écran secondaire. Ce mécanisme permet d’analyser le réseau étudié à deux niveaux de granularité différents : à l’échelle du graphe et à l’échelle d’une communauté.

vée pour pouvoir détecter les “swipes”. Dans l’état de sélection, il détecte également les clics.

7.3.3 Interactions multimodales

L’agrégation hiérarchique des nœuds est une composante essentielle de la représentation macroscopique des graphes complexes. Mais à ce niveau de granularité, une partie de l’information n’est pas visible. Ce problème de *focus & context* peut être partiellement endigué en utilisant différentes vues sur les données [15]. Une approche classique est alors d’afficher un aperçu de l’intégralité du graphe dans une fenêtre séparée pendant que l’utilisateur interagit avec la restitution actuelle. Cette approche est notamment utilisée dans le logiciel Tulip [9]. Cette approche peut comporter le risque de surcharger l’écran ; ce qui peut altérer la fouille du graphe. Nous avons opté pour une alternative multimodale : l’utilisateur peut transmettre une partie du graphe à un écran secondaire (par exemple une tablette tactile) équipée d’une version allégée de *TempoSpring*. Sans utiliser la bibliothèque d’interaction immersive, l’utilisateur peut transmettre le méta-nœud sélectionné en appuyant à la fois sur le bouton gauche de la souris et sur un raccourci clavier. Lorsqu’elle est utilisée, il suffit d’effectuer un “clic” sur le méta-nœud puis un “swipe” vers la gauche (un “swipe vers la droite” permettant de le désagréger). Cette stratégie permet des interactions complémentaires en parallèle à différentes échelles.

A l’heure actuelle, la version allégée de *TempoSpring* n’est disponible que sur le système d’exploitation mobile *Android* et nécessite un appareil doté d’un écran tactile. Pour assurer la cohérence de la restitution, cette version ne comprend pas

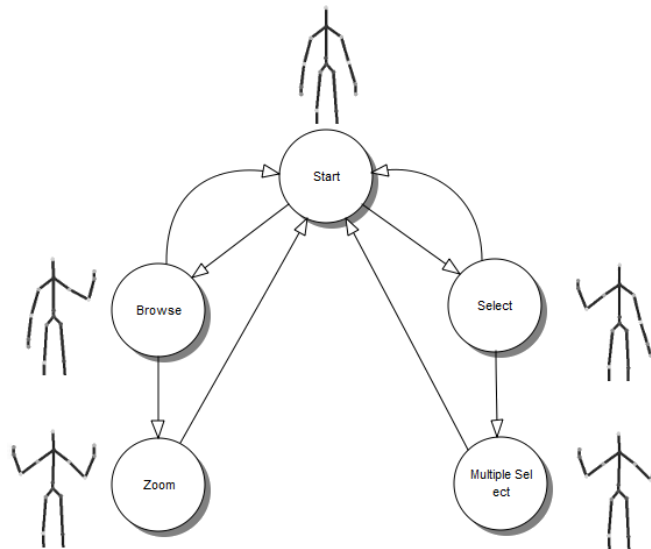


FIGURE 7.9 – Diagramme de l'automate du moteur d'interactions. Dans les états d'exploration et de sélection, le moteur d'interactions suit activement la vitesse de déplacement de la main qui n'est pas levée pour pouvoir détecter les “swipes”. dans l'état de sélection, il détecte également les “clics” de la main.

d'algorithme spécifique de tracé : elle affiche les nœuds et les liens aux même positions et avec les mêmes couleurs que sur l'écran principal. Il est alors possible d'interagir avec cette restitution. Toucher un nœud permet d'en afficher les informations sémantiques sous forme de fenêtre *pop-up*. En faisant glisser son doigt sur l'écran, l'utilisateur peut également appliquer une rotation sur le graphe. Le geste “pincer pour zoomer” est également pris en charge. Notons que les interactions sur les deux restitutions sont indépendantes : effectuer un zoom sur la version allégée n'entraîne pas de zoom sur l'écran principal.

Conclusion

Très récemment, Reda et al. ont présenté leur nouvel environnement immersif de visualisation : le CAVE2 [163]. Doté de 72 écrans LCD arrangés pour former un cylindre de 7.3m de diamètre, il offre une vue panoramique de 320 degrés avec une résolution de 72Mpixels au centre de laquelle un groupe de personnes peut travailler confortablement. Ce type d'environnement est paradigmatique des besoins de la fouille visuelle d'aujourd'hui : l'évolution des capacités de restitution des écrans classiques ne suit pas l'évolution de la taille des données à visualiser.

Dans un tel contexte, un autre paradigme d'interaction que l'habituel clavier-souris-écran est nécessaire pour que les différents utilisateurs puissent analyser les données. Dans ce chapitre, nous avons présenté une contribution pour tenter de pallier ce problème sous la forme d'une bibliothèque utilisant le capteur *Kinect* pour détecter des gestes déclencheurs d'interactions. Cette bibliothèque a été intégrée à

l'environnement *TempoSpring* pour permettre la fouille visuelle de graphes debout devant un écran de grande taille. Nous avons également présenté les capacités multimodales de *TempoSpring* qui offrent aux utilisateurs la possibilité de visualiser en simultané différents niveaux de granularité d'un graphe.

Des travaux conséquents seraient évidemment nécessaires pour proposer une solution fonctionnelle permettant, comme dans le CAVE2, des mécanismes collaboratifs facilitant l'interaction entre plusieurs utilisateurs.



8

Conclusion et perspectives

The more I learn, the more I learn how little I know.
—Socrates

Sommaire

8.1 Conclusion	120
8.2 Perspectives	122

8.1 Conclusion

L'*écran ultime* de Sutherland n'est pas encore une réalité, mais des technologies immersives reposant sur des images virtuelles en 3D se développent dans différents secteurs technologiques. Dans certaines applications, il s'agit de simuler la réalité perçue ; dans d'autres, la 3D permet d'afficher une quantité d'informations supérieure aux restitutions bi-dimensionnelles classiques ou de respecter la structure spatiale des objets restitués. Pourtant, la 3D peine à convaincre dans la communauté de fouille visuelle. Les raisons peuvent être multiples : une certaine méconnaissance des mécanismes psychophysiques impliqués dans la perception de la profondeur, des performances non optimales des matériels utilisés ou des modèles de restitution non adaptés.

Nos recherches ravivent le débat sur l'usage de la 3D pour des tâches de fouille visuelle sur des objets combinatoires (ici des graphes). Pour faire des comparaisons nous avons développé un protocole expérimental en nous restreignant à un modèle particulier de tracé. Dans le cas de la visualisation en 2D des approches plus performantes ont été proposées (e.g. [147]). Cependant, leur transposition à la 3D nécessiterait des changements profonds dans les restitutions et pose des questions ouvertes. Par conséquent, la construction d'un protocole permettant une comparaison entre des restitutions mettant en jeu des méthodes de tracé différentes pour les différents supports est une problématique qui nous paraît encore largement ouverte au débat.

Dans le cadre restreint que nous avons choisi d'approfondir, celui de la comparaison entre la 3D monoscopique et la 3D stéréoscopique, nos résultats expérimentaux concluent sur un intérêt de la stéréoscopie pour une tâche de détection de structures topologiques (les communautés) sur des graphes construits spécifiquement pour l'expérimentation. Les analyses doivent se prolonger sur des graphes de terrain. Nous avons déjà exploré l'intérêt de la 3D stéréoscopique dans le cadre applicatif du projet ANR GRAPHCOMP. Ce projet, pour lequel l'équipe de recherche dans laquelle j'ai effectué ma thèse était responsable local, consistait à explorer des graphes de relations sociales construits à partir d'archives de transactions agraires recueillies sur toute la période du Moyen Age dans une zone restreinte du Sud-Ouest de la France. Ces archives concernaient des transactions entre paysans et l'objectif pour les historiens était de mieux appréhender les relations qui avaient pu se nouer. Cette population qui représentait plus de 90% de la population de l'époque avait été longtemps négligée dans les recherches à cause du prisme déformant de la documentation qui concerne principalement les couches dominantes (le clergé et la noblesse). A partir des archives nous avons pu construire des graphes dont

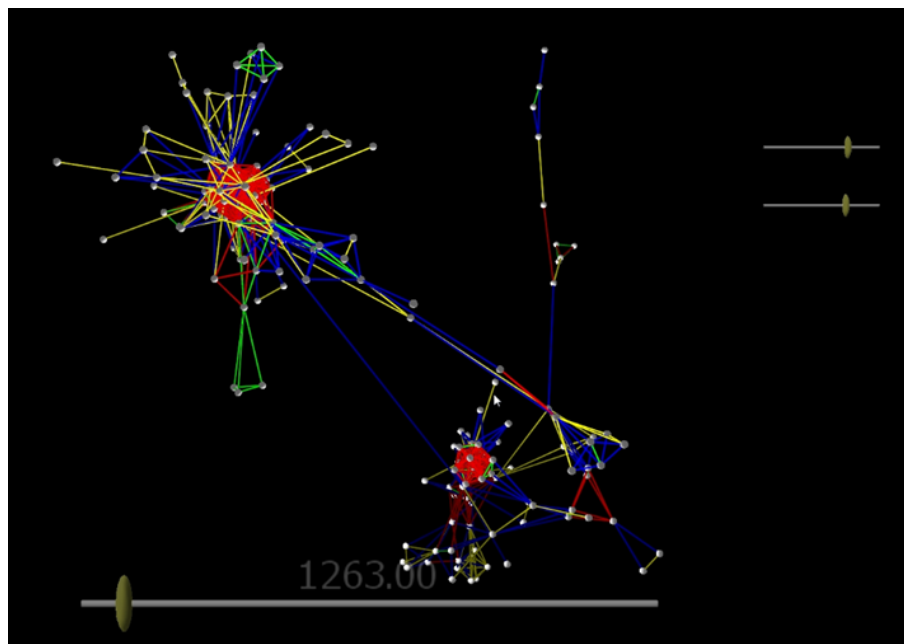


FIGURE 8.1 – Illustration d’un graphe du projet ANR GRAPHCOMP. Sur cette capture, nous pouvons distinguer deux communautés distinctes (à gauche et à droite) qui sont reliées par des individus “relais”. Cet instantané du réseau social correspond à l’année 1263.

les nœuds sont des noms inscrits sur les archives et les liens modélisent la présence conjointe de deux noms sur une même transaction (laissant supposer une relation sociale). Nous avons utilisé l’environnement *TempoSpring* pour représenter ces graphes et leur évolution et permettre à l’historien qui pilotait l’analyse une exploration *in situ* à des échelles qu’il n’avait jamais manipulées jusqu’alors. Ses retours ont été très positifs et ont permis de commencer à déceler des structurations intéressantes (communautés, individus “relais” entre des communautés différentes, évolution de familles importantes, ... voir la figure 8.1) qui ont ensuite été approfondies par des approches métrologiques. Mais, comme nous étions sur un terrain très novateur pour l’équipe nous n’avions pas structuré préalablement les phases d’entretiens avec l’historien expert des données et d’un point de vue scientifique les retours qu’il nous a donnés sont peu exploitables.

Au-delà de l’analyse des performances de la tâche, l’analyse d’un processus de fouille doit intégrer la prise en compte de l’interaction. Nous nous sommes restreints ici à une interaction simple qui permet à l’utilisateur d’appliquer une rotation à son point de vue autour du graphe. Et cette simplicité nous a cependant permis de distinguer différents types de comportements selon les modes de restitution. Pour mieux comprendre l’origine de ces différences nous nous sommes rapprochés de neurologues du centre de recherche Cerveau et Cognition de l’université de Toulouse, spécialistes de la perception de l’espace et du contexte 3D. Ils ont été très

intéressés par nos observations et nous ont confirmé que nos questions étaient encore très ouvertes, en particulier pour des tâches de haut niveau telles que celles qui nous concernent en fouille de données. Nous pensons que l'analyse de l'interaction est absolument essentielle dans les systèmes de fouille qui visent justement à mettre l'utilisateur au centre du processus.

8.2 Perspectives

Les recherches présentées dans ce manuscrit se placent dans le cadre de la fouille visuelle de données qui, bien que les problématiques sous-jacentes ne soient pas nouvelles, a connu un essor cette dernière décennie, porté à la fois par les besoins suscités par les changements d'échelles des données à traiter, et les évolutions technologiques qui permettent de développer de nouvelles interfaces visuelles interactives. Dans ce contexte, nous avons investigué l'apport d'une technologie (la visualisation stéréoscopique) dont le déploiement hors d'un cercle très restreint est tout récent. Par conséquent, les perspectives sont très nombreuses et concernent des aspects méthodologiques et technologiques pour le développement de nouvelles restitutions interactives bien adaptées à la complexité croissante des données mais aussi expérimentaux pour évaluer leur apport spécifique eu égard aux environnements existants. Dans la suite, nous ne développons pas l'ensemble de ces sujets que nous avons évoqués ponctuellement dans les différents chapitres et nous nous focalisons sur des perspectives à court terme qui sont déjà en court de développement.

Vers de nouveaux modes d'interaction Des travaux préconisent l'utilisation de la 3D stéréoscopique sur de grands écrans avec par exemple, un projecteur. Dans ce contexte de visualisation, l'utilisateur se trouve généralement debout, à plusieurs mètres de l'écran et il devient alors difficile d'utiliser les périphériques d'interaction classiques. Dans cette optique, nous avons implémenté une bibliothèque d'interaction "mains libres" permettant d'utiliser le capteur *Kinect* de *Microsoft* au sein d'un logiciel. Cette bibliothèque est notamment utilisée par *TempoSpring* pour offrir des interactions plus adaptées à une visualisation en station debout devant un grand écran. Mais cette approche reste à valider expérimentalement pour s'assurer qu'elle offre les mêmes possibilités que le duo souris/clavier dans une tâche de fouille visuelle tout en restant ergonomique (ex. : faire attention à la fatigue ou aux douleurs musculaires).

Vers une intégration de la dynamique Comme nous l'avons mentionné dans le chapitre 4, les graphes dynamiques sont devenus ces dernières années un su-

jet d'étude en plein essor. Il soulève de nombreuses problématiques à la fois pour l'analyse (ex. : comment catégoriser une communauté dont les membres peuvent disparaître et apparaître avec le temps ?) et la visualisation (ex. : comment souligner les changements dans le tracé sans perturber les parties qui restent stables ?).

En visualisation de graphes dynamiques, il est souvent primordial de préserver la carte mentale de l'utilisateur lors de restitutions successives. Or, certains commentaires évoqués durant nos expériences semblent indiquer qu'il existe des similitudes entre ce problème et des retours d'expérimentation que nous avons notés. En effet, des participants ont déclaré qu'il était difficile de "suivre" les communautés identifiées quand elles passaient à l'arrière plan à la suite des interactions. Ce phénomène ne nous a été reporté que sous la condition monoscopique. Nous envisageons donc d'étudier l'impact de la stéréoscopie sur la préservation de la carte mentale dans le cadre d'une visualisation de graphes dynamiques.

Vers une intégration de l'information spatiale Les données du projet GRAPHCOMP que nous avons évoqué ci-dessus comportent, en sus de l'information combinatoire, des informations spatiales : chaque individu est attaché à un lieu géolocalisé avec des précisions très variables. Il est donc possible d'étudier les relations entre proximité spatiale et proximité sociale. Une partie des liens sociaux s'expliquent-ils par une proximité spatiale ? Cette question suscite l'intérêt de la communauté d'analyse des réseaux sociaux [11, 129]. Pour y contribuer, nous avons commencé à explorer deux pistes : (i) l'ajout d'une représentation de la zone géographique au tracé d'un graphe pour mieux saisir les contraintes géographiques régissant les connexions entre individus (ii) une déformation du tracé du graphe en fonction de l'information spatiale pour aider à l'identification de connexions "non triviales" (individus éloignés mais connectés, individus proches mais non connectés, individus reliant deux communautés géolocalisées, etc ...).

L'usage de la 3D est particulièrement adapté pour le premier point : dans notre cas, nous ajoutons un fond de carte en relief de la région correspondant au graphe (une illustration est fournie sur la figure 8.2). L'hypothèse sous-jacente validée par les historiens est que, pour la zone géographique considérée, le relief a peu changé depuis la période des données. En 3D, les caractéristiques géographiques comme les collines, montagnes ou lits de rivières sont intelligibles sans codage préalable et peuvent contribuer à expliquer l'existence ou l'absence de certaines connexions. Par exemple, deux communautés appartenant à deux lieux différents peuvent avoir très peu de liens si ces lieux sont séparés par une rivière ou une colline.

En complément, une déformation du tracé du graphe selon les proximités géographiques peut faciliter la découverte des relations entre la combinatoire des lieux et les plongements spatiaux des nœuds. Nous avons ainsi raffiné l'algorithme de forces en y ajoutant une force gravitationnelle. Chacun des lieux apparaissant dans

la base de données agit comme une source d'attraction pour tous les nœuds représentant les individus qui lui sont rattachés. L'idée sous-jacente étant qu'une fois la convergence atteinte, il est plus facile d'identifier les individus reliant différents lieux ou les individus isolés pour un lieu donné. Dans l'implémentation actuelle, cette force peut être activée et désactivée, ce qui entraîne une déformation du tracé en temps réel. Cette déformation permet d'avoir une première intuition de la façon dont les caractéristiques géographiques impactent le réseau : si le tracé est très peu déformé cela peut signifier que les contraintes sont fortes, tandis qu'une déformation importante suggère une absence de causalité entre les liens sociaux et certaines caractéristiques géographiques du terrain de l'étude.

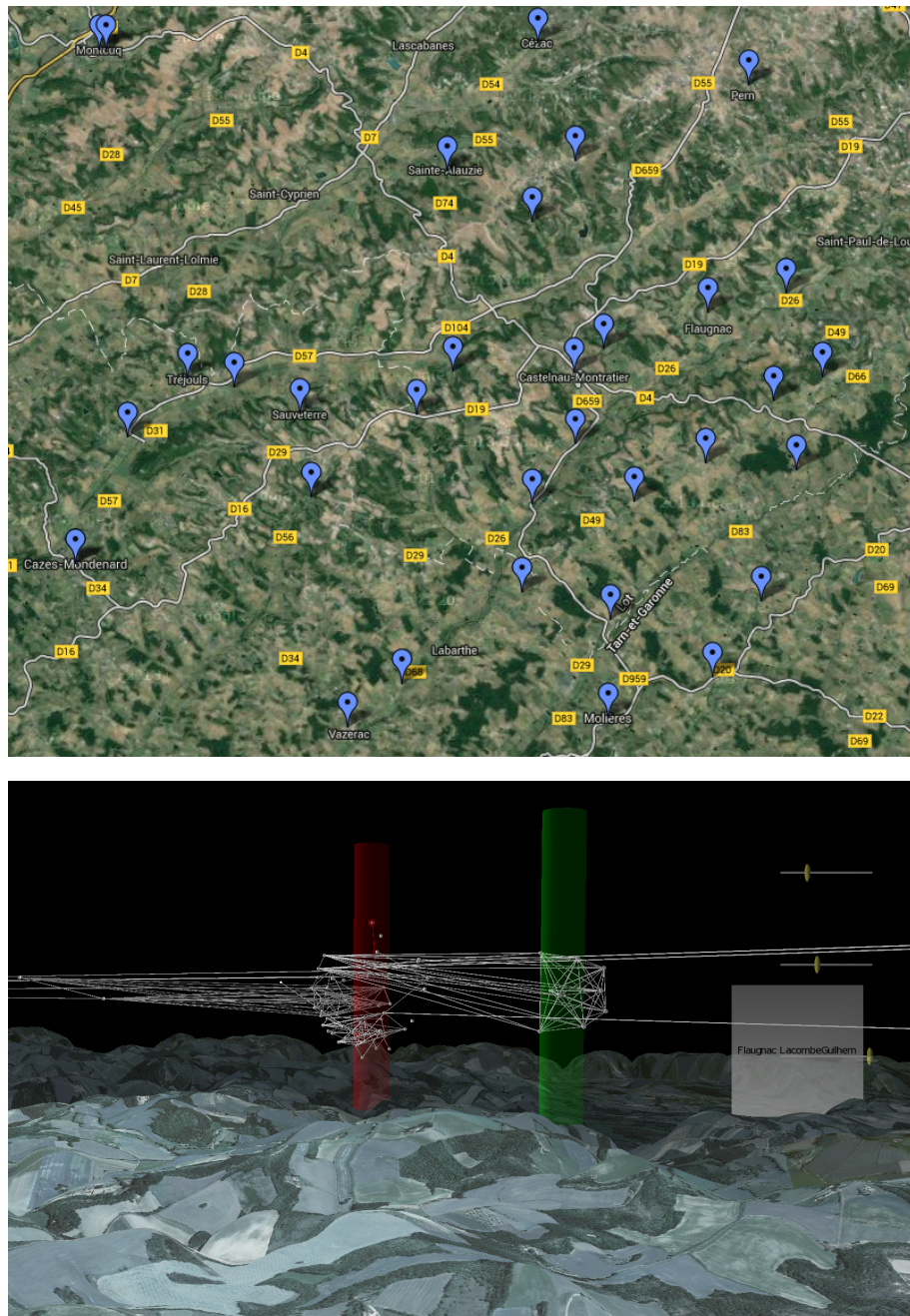


FIGURE 8.2 – Illustration du plongement spatial dans *TempoSpring*. Sur la première capture nous pouvons observer la géolocalisation des lieux auxquels sont attachés les individus du graphe du projet GRAPHCOMP. Sur la deuxième capture, une partie de ce graphe est juxtaposée à un fond de carte de la région correspondante qui est déformé en fonction du relief. Le fond de carte et les données relatives au relief proviennent des bases de données d’IGN. Deux nœuds sont sélectionnés et la localisation des lieux auxquels les individus qu’ils représentent sont rattachés est représentée par un cylindre translucide (teinte rouge pour le premier individu et teinte verte pour le second). Lorsque le tracé n’est pas déformé, cette fonctionnalité permet de mettre en parallèle la proximité topologique dans le réseau et la proximité géographique sur le fond de carte.

Bibliographie

- [1] Accot, J., Zhai, S. : Beyond fitts' law : models for trajectory-based hci tasks. In : Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '97, pp. 295–302. ACM, New York, NY, USA (1997). DOI <http://doi.acm.org/10.1145/258549.258760>. URL <http://doi.acm.org/10.1145/258549.258760> 91
- [2] Ahn, J., Taieb-Maimon, M., Sopan, A., Plaisant, C., Shneiderman, B. : Temporal visualization of social network dynamics : Prototypes for nation of neighbors. In : International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction (SBP11) (2011) 71
- [3] Alper, B., Hollerer, T., Kuchera-Morin, J., Forbes, A. : Stereoscopic highlighting : 2d graph visualization on stereo displays. IEEE Transactions on Visualization and Computer Graphics **17**(12), 2325–2333 (2011) 38, 39
- [4] Amar, R., Eagan, J., Stasko, J. : Low-level components of analytic activity in information visualization. In : Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on, pp. 111–117. IEEE (2005) 9, 43, 44
- [5] Archambault, D., Munzner, T., Auber, D. : Topolayout : Multi-level graph layout by topological features. IEEE Ttrans. on Visualization and Computer Graphics **13** (2007) 15, 53
- [6] Archambault, D., Purchase, H., Pinaud, B. : Animation, Small Multiples, and the Effect of Mental Map Preservation in Dynamic Graphs. IEEE Transactions on Visualization and Computer Graphics **17**, 539–552 (2011). URL <http://hal.inria.fr/inria-00472423/en/> 72
- [7] Archambault, D., Purchase, H., Pinaud, B. : Difference Map Readability for Dynamic Graphs. In : U. Brandes, S. Cornelsen (eds.) 18th International Symposium on Graph Drawing 18th International Symposium on Graph Drawing, LNCS, vol. 6502, pp. 50–61. Springer, Constance Allemagne (2011) 71
- [8] Arditi, A. : Handbook of perception and human performance, chap. Binocular vision, pp. 23 :1 – 23 :41. New York : Wiley (1986) 25

- [9] Auber, D. : Tulip : A huge graph visualisation framework. In : P. Mutzel, M. Jünger (eds.) Graph Drawing Softwares, Mathematics and Visualization, pp. 105–126. Springer-Verlag (2003) 48, 116
- [10] Auber, D., Chiricota, Y., Delest, M., Melançon, G., Domenger, J.P., Mary, P. : Visualisation de graphes avec Tulip : exploration interactive de grandes masses de données en appui à la fouille de données et à l'extraction de connaissances. In : G.V. Monique Noirhomme Fraiture (ed.) EGC'07 : Extraction et Gestion de Connaissances, pp. 147–156. Cépaduès, Namur, Belgique (2007). URL <http://hal-lirmm.ccsd.cnrs.fr/lirmm-00203728/en/> 38, 61
- [11] Backstrom, L., Sun, E., Marlow, C. : Find me if you can : improving geographical prediction with social and spatial proximity. In : Proceedings of the 19th international conference on World wide web, WWW '10, pp. 61–70. ACM, New York, NY, USA (2010). DOI 10.1145/1772690.1772698. URL <http://doi.acm.org/10.1145/1772690.1772698> 123
- [12] Barnes, J., Hut, P. : A hierarchical $O(n \log n)$ force-calculation algorithm (1986) 61
- [13] Bastian, M., Heymann, S., Jacomy, M. : Gephi : An open source software for exploring and manipulating networks. In : ICWSM'09 (2009) 38
- [14] Battista, G.D., Patrignani, M., Vargiu, F. : A split&push approach to 3d orthogonal drawing. In : Proceedings of the 6th International Symposium on Graph Drawing, GD '98, pp. 87–101. Springer-Verlag, London, UK (1998). URL <http://portal.acm.org/citation.cfm?id=647550.729089> 51
- [15] Baudisch, P., Good, N., Stewart, P. : Focus plus context screens : combining display technology with visualization techniques. In : Proceedings of the 14th annual ACM symposium on User interface software and technology, UIST '01, pp. 31–40. ACM, New York, NY, USA (2001). DOI 10.1145/502348.502354. URL <http://doi.acm.org/10.1145/502348.502354> 53, 116
- [16] Baur, M. : Software for the analysis and visualization of social networks. Ph.D. thesis, Karlsruhe University (2008) 49
- [17] Baur, M., Schank, T. : Dynamic graph drawing in visone. Tech. rep. (2008) 71
- [18] Beaudouin-Lafon, M. : Designing interaction, not interfaces. In : Proceedings of the working conference on Advanced visual interfaces, AVI '04, pp. 15–22. ACM, New York, NY, USA (2004). DOI 10.1145/989863.989865. URL <http://doi.acm.org.gate6.inist.fr/10.1145/989863.989865> 108

- [19] Belcher, D., Billingham, M., Hayes, S., Stiles, R. : Using augmented reality for visualizing complex graphs in three dimensions. In : ISMAR, pp. 84–92 (2003) 39
- [20] Bender-deMoll, S., McFarland, D. : The Art and Science of Dynamic Network Visualization. JoSS : Journal of Social Structure 7 (2005). URL <http://www.cmu.edu/joss/content/articles/volume7/deMollMcFarland/> 68, 71
- [21] Bernardino, A., Santos-Victor, J., Ferre, M., Sanchez-Urán, M. : Stereoscopic image visualization for telerobotics. experiments with active binocular cameras. In : M. Ferre, M. Buss, R. Aracil, C. Melchiorri, C. Balaguer (eds.) *Advances in Telerobotics, Springer Tracts in Advanced Robotics*, vol. 31, pp. 77–90. Springer Berlin Heidelberg (2007). DOI 10.1007/978-3-540-71364-7_6. URL http://dx.doi.org/10.1007/978-3-540-71364-7_6 35
- [22] Beurden, M., Ijsselstein, W., Juola, J. : Effectiveness of stereoscopic displays in medicine : A review. 3D Research 3(1), 1–13 (2012) 37
- [23] van Beurden, M.H., Ijsselstein, W.A., de Kort, Y.A. : Evaluating stereoscopic displays : both efficiency measures and perceived workload sensitive to manipulations in binocular disparity pp. 786,316–786,316–7 (2011). DOI 10.1117/12.872566. URL <http://dx.doi.org/10.1117/12.872566> 37
- [24] Biedl, T., Kant, G. : A better heuristic for orthogonal graph drawings. In : J. van Leeuwen (ed.) *Algorithms - ESA '94, Lecture Notes in Computer Science*, vol. 855, pp. 24–35. Springer Berlin / Heidelberg (1994). URL <http://dx.doi.org/10.1007/BFb0049394>. 10.1007/BFb0049394 51
- [25] Bingham, G.P. : Perceiving the size of trees : Form as information about scale. *Journal of experimental psychology. Human perception and performance* 19(6), 1139–1161 (1993) 28
- [26] Bonferroni, C.E. : *Teoria statistica delle classi e calcolo delle probabilita*. Libreria internazionale Seeber (1936) 87
- [27] Bowman, D.A., McMahan, R. : Virtual reality : how much immersion is enough ? *Computer* 40 (7), 36–43 (2007) 22
- [28] Brandes, U., Dwyer, T., Schreiber, F. : Visualizing related metabolic pathways in two and a half dimensions. In : G. Liotta (ed.) *Graph Drawing, Lecture Notes in Computer Science*, vol. 2912, pp. 111–122. Springer Berlin / Heidelberg (2004) 38

- [29] Brandes, U., Wagner, D. : Visone - analysis and visualization of social networks. In : Graph Drawing Software, pp. 321–340. Springer-Verlag (2003) [54](#)
- [30] Braunstein, M. : Depth perception through motion. New York : Academic Press (1976) [30](#)
- [31] Bressand, A., Nicolaidis, K. : Les services au coeur de l'économie relationnelle. Revue d'économie industrielle **43**(1), 141–163 (1988) [14](#)
- [32] Brooks, F.P. : What's real about virtual reality ? IEEE Computer Graphics and Applications **19**(6), 16–27 (1999). DOI 10.1109/38.799723. URL <http://dx.doi.org/10.1109/38.799723> [22](#)
- [33] Broy, N., André, E., Schmidt, A. : Is stereoscopic 3d a better choice for information representation in the car? In : Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications, AutomotiveUI '12, pp. 93–100. ACM, New York, NY, USA (2012). DOI 10.1145/2390256.2390270. URL <http://doi.acm.org/10.1145/2390256.2390270> [35](#)
- [34] Bullmore, E., Sporns, O. : Complex brain networks : graph theoretical analysis of structural and functional systems. Nature Reviews Neuroscience **10**(3), 186–198 (2009) [14](#)
- [35] Burton, H. : The optics of euclid. In : Journal of the Optical Society of America **35**, 5, pp. 357–372. Springer (1945) [16](#)
- [36] Cazabet, R., Amblard, F., Hanachi, C. : Detection of overlapping communities in dynamical social networks. In : Social Computing (SocialCom), 2010 IEEE Second International Conference on, pp. 309–314. IEEE (2010) [46](#)
- [37] Chan, A., Lau, R.W.H., Ng, B. : Motion prediction for caching and prefetching in mouse-driven dve navigation. ACM Transactions on Internet Technologies **5**, 70–91 (2005). DOI <http://doi.acm.org/10.1145/1052934.1052937>. URL <http://doi.acm.org/10.1145/1052934.1052937> [93](#)
- [38] Chen, J., Cai, H., Auchus, A.P., Laidlaw, D.H. : Effects of stereo and screen size on the legibility of three-dimensional streamtube visualization. IEEE transactions on Visualization and Computer Graphics **18** (12) (2012) [36](#)
- [39] Chen, Y., Liestman, A., Liu, J. : Clustering algorithms for ad hoc wireless networks. Ad Hoc and Sensor Networks **28** (2004) [76](#)
- [40] Cho, I., Dou, W., Wartell, Z., Ribarsky, W., Wang, X. : Evaluating depth perception of volumetric data in semi-immersive vr. In : Proceedings of the International Working Conference on Advanced Visual Interfaces,

- AVI '12, pp. 266–269. ACM, New York, NY, USA (2012). DOI 10.1145/2254556.2254606. URL <http://doi.acm.org/10.1145/2254556.2254606> 35
- [41] Cockburn, A., McKenzie, B. : 3d or not 3d ? : evaluating the effect of the third dimension in a document management system. In : Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '01, pp. 434–441. ACM, New York, NY, USA (2001). DOI 10.1145/365024.365309. URL <http://doi.acm.org/10.1145/365024.365309> 16, 32
- [42] Cohen, J.D. : Drawing graphs to convey proximity : an incremental arrangement method. ACM Trans. Comput.-Hum. Interact. **4**(3), 197–229 (1997). DOI 10.1145/264645.264657. URL <http://doi.acm.org/10.1145/264645.264657> 50, 51
- [43] Cole, R.E., Merritt, J.O., Fore, S., Lester, P. : Remote-manipulator tasks impossible without stereo tv. In : Proc. SPIE Vol. 1256, p. 255-265, Stereoscopic Displays and Applications, Scott S. Fisher ; John O. Merritt ; Eds (1990) 35
- [44] Collberg, C., Kobourov, S., Nagra, J., Pitts, J., Wampler, K. : A system for graph-based visualization of the evolution of software. In : In Proceedings of the 2003 ACM symposium on Software visualization, pp. 77–86. ACM Press (2003) 71
- [45] Condon, A., Karp, M. : Algorithms for graph partitioning on the planted partition model. Random Structures and Algorithms **18**, 116–140 (2001) 79
- [46] Corinilleau-Pérès, V., Droulez, J. : The visual perception of three-dimensional shape from self-motion and object-motion. In : Vision Research, vol. 34, pp. 2331–2336 (1994) 77
- [47] Cruz, I.F., Twarog, J.P. : 3d graph drawing with simulated annealing. In : Graph Drawing, pp. 162–165. Springer (1996) 51
- [48] Cugini, J., Scholtz, J. : Visvip : 3d visualization of paths through web sites. In : Proceedings of the 10th International Workshop on Database & Expert Systems Applications, DEXA '99, pp. 259–. IEEE Computer Society, Washington, DC, USA (1999). URL <http://portal.acm.org/citation.cfm?id=519627.790354> 38
- [49] Cutting, J. : How the eye measures reality and virtual reality. Behavior Research Methods, Instrumentation, and Computers **29**, 29–36 (1997) 23, 29, 30, 31
- [50] Cutting, J.E., Vishton, P.M. : Perception of space and motion, chap. Perceiving layout and knowing distances : The integration, relative potency, and

- contextual use of different information about depth, pp. 69–117. San Diego : Academic Press (1995) [26](#)
- [51] Deligiannidis, L., Sheth, A.P., Aleman-meza, B. : Semantic analytics visualization. In : In Proc. of the IEEE International Conference on Intelligence and Security Informatics (ISI-2006) (2006) [38](#)
- [52] Di-Battista, G., Eades, P., Tamassia, R., Tollis, I. : Graph Drawing - Algorithms for the visualization of graphs. Prentice Hall (1999) [15](#), [49](#)
- [53] Diehl, S., Görg, C. : Graphs, they are changing - dynamic graph drawing for a sequence of graphs. In : Proc. 10th Int. Symp. Graph Drawing (GD 2002), number 2528 in Lecture Notes in Computer Science, LNCS, pp. 23–31. Springer-Verlag (2002) [72](#)
- [54] Doi, A., Koide, A. : An efficient method of triangulating equivalued surfaces by using tetrahedral cells. IEICE Transactions Communication, Electronics, Information and Systems E74(1) pp. 214–224 (1991) [33](#)
- [55] Domini, F., Caudek, C., Tassinari, H. : Stereo and motion information are not independently processed by the visual system. In : Vision Research, vol. 46, pp. 1707–23. Elsevier (2006) [31](#)
- [56] Doreian, P. : On the evolution of group and network structure ii : Structures within structure. Social Networks **8**(1), 33 – 64 (1986) [69](#)
- [57] Drascic, D., Milgram, P., Grodski, J. : Learning effects in telemanipulation with monoscopic versus stereoscopic remote viewing. In : Systems, Man and Cybernetics, 1989. Conference Proceedings., IEEE International Conference on, pp. 1244–1249 vol.3 (1989). DOI 10.1109/ICSMC.1989.71502 [37](#)
- [58] Durgin, F.H. : Texture density adaptation and the perceived numerosity and distribution of texture. Journal of Experimental Psychology : Human Perception & Performance **21**, 141–169 (1995) [28](#)
- [59] Dynes, S.B.C., Gloor, P.A., Gloor, P.A., Gloor, P.A., Laubacher, R., Laubacher, R., Zhao, Y., Zhao, Y., Dynes, S. : Temporal visualization and analysis of social networks. In : NAACSOS Conference, June 27 - 29, Pittsburgh PA, North American Association for Computational Social and Organizational Science. In (2004) [71](#)
- [60] Eades, P. : A heuristic for graph drawing. In : Congressus Numerantium, 42, pp. 149–160 (1984) [50](#), [51](#)
- [61] Eades, P. : On the future of graph drawing. In : Invited talk, 18th International Symposium on Graph Drawing (2010) [15](#), [16](#)
- [62] Eades, P., Lai, W., Misue, K., Sugiyama, K. : Preserving the mental map of a diagram. International Institute for Advanced Study of Social Information Science, Fujitsu Limited (1991) [71](#)

- [63] Eades, P., Lin, X. : Spring algorithms and symmetry. *Theoretical Computer Science* **240**, 379–405 (1999) [51](#)
- [64] Eades, P., Stirk, C., Whitesides, S. : The techniques of komolgorov and bardzin for three-dimensional orthogonal graph drawings. *Information Processing Letters* **60**(2), 97–103 (1996) [49](#), [51](#)
- [65] Eades, P., Symvonis, A., Whitesides, S. : Three-dimensional orthogonal graph drawing algorithms. *Discrete Applied Mathematics* **103**(1-3), 55–87 (2000) [38](#)
- [66] Ekman, G. : Weber’s law and related functions. *The Journal of Psychology* **47**(2), 343–352 (1959) [30](#)
- [67] Falkowski, T., Bartelheimer, J., Spiliopoulou, M. : Mining and visualizing the evolution of subgroups in social networks. In : *WI ’06 : Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 52–58. IEEE Computer Society (2006) [69](#), [70](#)
- [68] Fekete, J.D., Wang, D., Dang, N., Aris, A., Plaisant, C. : Overlaying graph links on treemaps. In : *IEEE Symposium on Information Visualization Conference Compendium (demonstration)*, vol. 5 (2003) [47](#)
- [69] Fekete, S., Houle, M., Whitesides, S. : New results on a visibility representation of graphs in 3d. In : F. Brandenburg (ed.) *Graph Drawing, Lecture Notes in Computer Science*, vol. 1027, pp. 234–241. Springer Berlin / Heidelberg (1996). URL <http://dx.doi.org/10.1007/BFb0021807>. [10.1007/BFb0021807](http://dx.doi.org/10.1007/BFb0021807) [52](#)
- [70] Férey, N., Gros, P.E., Hérisson, J., Gherbi, R. : Visual data mining of genomic databases by immersive graph-based exploration. In : *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia, GRAPHITE ’05*, pp. 143–146. ACM, New York, NY, USA (2005) [40](#)
- [71] Fernanda, B., Viégas, Donath, J. : Social network visualization : can we go beyond the graph. In : *Workshop on Social Networks for Design and Analysis : Using Network Information in CSCW* (2004, pp. 6–10 (2004) [47](#)
- [72] Ferre, M., Aracil, R., Sanchez-Uran, M. : Stereoscopic human interfaces. *Robotics Automation Magazine, IEEE* **15**(4), 50–57 (2008). DOI [10.1109/MRA.2008.929929](https://doi.org/10.1109/MRA.2008.929929) [37](#)
- [73] Fitts, P. : The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* **47**, 381–391 (1954) [91](#)
- [74] Fortunato, S. : Community detection in graphs. *Physics Reports* **486**(3), 75–174 (2010) [16](#), [46](#), [82](#)

- [75] Freeman, L. : Visualizing social networks. In : Journal of Social Structure, vol. 1 (2000) [47](#)
- [76] Friedman, M. : The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association **32**(200), 675–701 (1937) [83](#)
- [77] Frishman, Y., Tal, A. : Online dynamic graph drawing. IEEE Transactions on Visualization and Computer Graphics **14**(4), 727–740 (2008) [61](#), [72](#)
- [78] Fruchterman, T.M.J., Reingold, E.M. : Graph drawing by force-directed placement. Software : Practice and Experiences **21**(11), 1129–1164 (1991) [50](#), [51](#), [58](#), [77](#)
- [79] keung Fung, W., Lo, W.T., Liu, Y., Xi, N. : A case study of 3d stereoscopic vs. 2d monoscopic tele-reality in real-time dexterous teleoperation. In : IROS, pp. 181–186 (2005) [35](#)
- [80] G., S., P., H.J. : Identification of pointing difficulties of two individuals with parkinson’s disease via a sub-movement analysis. In : Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems (CHI EA ’11), pp137-140 (2011) [93](#)
- [81] Gansner, E.R., Koren, Y., North, S. : Graph drawing by stress majorization. In : Graph Drawing, pp. 239–250. Springer (2005) [50](#)
- [82] Garbers, J., Prömel, H.J., Steger, A. : Finding clusters in vlsi circuits. In : Proceedings of ICCAD’90, pp. 520–523 (1990) [79](#)
- [83] Gateau, S., Neuman, R. : Stereoscopy from xy to z. In : Courses of SIGGRAPH Asia (2010) [63](#)
- [84] Gerhard, S., Cammoun, L., Thiran, J.P., Hagmann, P. : Connectomeviewer.org. ecole polytechnique fédérale de lausanne and university hospital center and university of lausanne (2010) [38](#)
- [85] Getty, D.J., Green, P.J. : Clinical applications for stereoscopic 3-d displays. Journal of the Society for Information Display **15**(6), 377–384 (2007) [34](#)
- [86] Ghoniem, M., Fekete, J.D., Castagliola, P. : On the readability of graphs using node-link and matrix-based representations : a controlled experiment and statistical analysis. Information Visualization **4**(2), 114–135 (2005) [47](#)
- [87] Gibson, J. : The ecological approach to visual perception (1986) [71](#)
- [88] Girvan, M., Newman, M.E. : Community structure in social and biological networks. Proceedings of the National Academy of Sciences **99**(12), 7821–7826 (2002) [46](#)
- [89] Golbeck, J., Hendler, J. : Filmtrust : Movie recommendations using trust in web-based social networks. In : Proceedings of the IEEE Consumer communications and networking conference, vol. 96. Citeseer (2006) [14](#)

- [90] Gombrich, E.H., Gombrich, E. : *Shadows : The depiction of cast shadows in Western art*. National Gallery Publications London (1995) 30
- [91] Greffard, N., Picarougne, F., Kuntz, P., et al. : *Librairie open source de visualisation interactive et immersive*. In : *Atelier Fouille Visuelle de Données : avancées récentes et perspectives, Extraction et Gestion des Connaissances (EGC'11)* (2011) 56
- [92] Halpin, H., Zielinski, D., Brady, R., Kelly, G. : *Exploring semantic social networks using virtual reality*. In : *Proceedings of the 7th International Conference on The Semantic Web, Lecture Notes In Computer Science*, vol. 5318, pp. 599–614 (2008) 39, 109
- [93] He, W., Marriott, K. : *Constrained graph layout*. In : *Graph Drawing*, pp. 217–232. Springer (1997) 50
- [94] Henry, N., daniel Fekete, J. : *Matlink : Enhanced matrix visualization for analyzing social networks*. In : *Proceedings of the International Conference Interact*, pp. 288–302 (2007) 44, 47
- [95] Hérisson, J., Gros, P.E., Férey, N., Magneau, O., Gherbi, R. : *Dna in virtuo visualization and exploration of 3d genomic structures*. In : *Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa, AFRIGRAPH '04*, pp. 35–40. ACM, New York, NY, USA (2004) 40
- [96] Herman, I., Melançon, G., Marshall, M.S. : *Graph visualization and navigation in information visualization : A survey*. *IEEE Transactions on Visualization and Computer Graphics* 6(1), 24–43 (2000). DOI 10.1109/2945.841119. URL <http://dx.doi.org/10.1109/2945.841119> 15, 49, 51, 52
- [97] Hernandez, A., Basset, O., Bremond, A., Magnin, I. : *Stereoscopic visualization of three-dimensional ultrasonic data applied to breast tumours*. *European Journal of Ultrasound* 8(1), 51–65 (1998) 34
- [98] Holliman, N. : *3d display systems*. Tech. rep., Department of Computer Science, Univ. Durham (2005) 33
- [99] Holten, D. : *Hierarchical edge bundles : Visualization of adjacency relations in hierarchical data*. *Visualization and Computer Graphics, IEEE Transactions on* 12(5), 741–748 (2006) 53
- [100] Hou, J., Nam, Y., Peng, W., Lee, K.M. : *Effects of screen size, viewing angle, and players' immersion tendencies on game experience*. *Computer in Human Behavior* 28(2), 617–623 (2012). DOI 10.1016/j.chb.2011.11.007. URL <http://dx.doi.org/10.1016/j.chb.2011.11.007> 111

- [101] Hsu, J., Babbs, C.F., Chelberg, D.M., Pizlo, Z. : Study of the effectiveness of stereo imaging with applications in mammography. *Proceedings of SPIE* **1913**, 154–165 (1993) 34
- [102] Huang, W., Hong, S.H., Eades, P. : Effects of sociogram drawing conventions and edge crossings in social network visualization. *Journal of Graph Algorithms and Applications* **11**(2), 397–429 (2007) 49
- [103] Hubona, G.S., Wheeler, P.N., Shirah, G.W., Brandt, M. : The relative contributions of stereo, lighting, and background scenes in promoting 3d depth visualization. *ACM Trans. Comput.-Hum. Interact.* **6**, 214–242 (1999). DOI <http://doi.acm.org/10.1145/329693.329695>. URL <http://doi.acm.org/10.1145/329693.329695> 31
- [104] Hughes, T., Hyun, Y., Liberles, D.A. : Visualising very large phylogenetic trees in three dimensional hyperbolic space. *BMC Bioinformatics* **5**, 48 (2004) 38
- [105] IJsselsteijn, W., Ridder, H.d., Freeman, J., Avons, S.E., Bouwhuis, D. : Effects of stereoscopic presentation, image motion, and screen size on subjective and objective corroborative measures of presence. *Presence : Teleoperators and Virtual Environments* **10**(3), 298–311 (2001). DOI 10.1162/105474601300343621. URL <http://dx.doi.org/10.1162/105474601300343621> 111
- [106] IJsselsteijn, W., Riva, G. : Being there : concepts, effects and measurements of user presence in synthetic environments, chap. Being There : The experience of presence in mediated environments, pp. 3–16. IOS Press (2003) 22
- [107] Iragne, F., Nikolski, M., Mathieu, B., Auber, D., Sherman, D. : Proviz : protein interaction visualization and exploration. *Bioinformatics* **21**(2), 272–274 (2005) 49
- [108] Johansson, G. : Monocular movement parallax and near-space perception. *Perception* **2**, 136–145 (1973) 26
- [109] Junger, M., Mutzel, P. : *Graph Drawing Software*. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2003) 15
- [110] Kamada, T., Kawai, S. : An algorithm for drawing general undirected graphs. *Information Processing Letters* **31**(1), 7–15 (1989) 50
- [111] Kang, H., Getoor, L., Singh, L. : Visual analysis of dynamic group membership in temporal social networks. *SIGKDD Explorations Newsletter* **9**, 13–21 (2007) 71
- [112] Kansky, K. : Structure of transportation networks : relationships between network geometry and regional characteristics. No. n° 84 in University of

- Chicago. Department of geography. University of Chicago. (1963). URL <http://books.google.fr/books?id=1HIgAQAAMAAJ> 14
- [113] Keim, D.A., Mansmann, F., Schneidewind, J., Thomas, J., Ziegler, H. : Visual data mining. chap. Visual Analytics : Scope and Challenges, pp. 76–90. Springer-Verlag, Berlin, Heidelberg (2008). DOI 10.1007/978-3-540-71080-6_6. URL http://dx.doi.org/10.1007/978-3-540-71080-6_6 15, 42, 43, 53
- [114] Kerr, B., Condon, S.M., McDonald, L.A. : Cognitive spatial processing and the regulation of posture. *Journal of Experimental Psychology : Human Perception and Performance* **11**(5), 617 (1985) 23
- [115] Kersten, M.A., Stewart, A.J., Troje, N., Ellis, R. : Enhancing depth perception in translucent volumes. *IEEE Transactions on Visualization and Computer Graphics* **12** (05), 1117–1123 (2006) 34
- [116] Kickuth, R., Hartung, G., Laufer, U., Gruening, C., Stueckle, C., Liermann, D. : Stereoscopic 3d ct vs. standard 3d ct in the classification of acetabular fractures : An experimental study. *British Journal of Radiology* **75**(893), 422–427 (2002) 36, 37
- [117] Kniss, J.M., Rezk-salama, C., Weiskopf, D., Engel, K. : Real-Time Volume Graphics. A.K. Peter (2006) 33
- [118] Knuth, D. : Computer draw flowcharts. *Communications of the ACM* **6**, 555–563 (1963) 43
- [119] Kolmogorov, A., Barzdin, Y. : About realization of sets in 3-dimensional space, problems in cybernetics. (1967) 51
- [120] Kruskal, W.H., Wallis, W.A. : Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association* **47**(260), 583–621 (1952) 87
- [121] Kubovy, M. : The psychology of perspective and Renaissance art. CUP Archive (1988) 29
- [122] Kumar, G., Garland, M. : Visual exploration of complex time-varying graphs. *IEEE Transactions on Visualization and Computer Graphics* **12**, 805–812 (2006) 71
- [123] Lalanne, L. : Essai d’une théorie des réseaux de chemins de fer, fondée sur l’observation des faits et sur les lois primordiales qui président au groupement des populations. *Comptes rendus des séances de l’Académie des Sciences* **57**(2), 206–210 (1863) 14
- [124] Landy, M.S., Maloney, L.T., Johnston, E.B., Young, M. : Measurement and modeling of depth cue combination. *Vision Research* **35**, 389–412 (1995) 25, 30

- [125] LaViola Jr., J.J., Prabhat, Forsberg, A.S., Laidlaw, D.H., van Dam, A. : Trends in Interactive Visualization, Advanced Information and Knowledge Processing, chap. Virtual Reality-Based Interactive Scientific Visualization Environments, pp. 225–250. Springer (2009) [22](#)
- [126] Lee, B., Plaisant, C., Parr, C.S., Fekete, J.D., Henry, N. : Task taxonomy for graph visualization. In : Proceedings of the 2006 AVI workshop on BEyond time and errors : novel evaluation methods for information visualization, BELIV '06, pp. 1–5. ACM, New York, NY, USA (2006). DOI 10.1145/1168149.1168168. URL <http://doi.acm.org/10.1145/1168149.1168168> [43](#), [44](#)
- [127] Leinhardt, S. : Social Networks : A Developing Paradigm. Quantitative Studies in Social Relations. Academic Press (1977). URL <http://books.google.fr/books?id=File3AAAAIAAJ> [14](#)
- [128] Lilliefors, H.W. : On the kolmogorov-smirnov test for normality with mean and variance unknown. Journal of the American Statistical Association **62**(318), 399–402 (1967). DOI 10.1080/01621459.1967.10482916. URL <http://amstat.tandfonline.com/doi/abs/10.1080/01621459.1967.10482916> [98](#)
- [129] Lima, A., Musolesi, M. : Spatial dissemination metrics for location-based social networks. In : Proceedings of the 2012 ACM Conference on Ubiquitous Computing, UbiComp '12, pp. 972–979. ACM, New York, NY, USA (2012). DOI 10.1145/2370216.2370429. URL <http://doi.acm.org/10.1145/2370216.2370429> [123](#)
- [130] Lorensen, W.E., Cline, H.E. : Marching cubes : A high resolution 3d surface construction algorithm. Computer Graphics **21**(4), 163–169 (1987) [33](#)
- [131] Loubier, E., Bahsoun, W., Dousset, B. : Visualization and analysis of large graphs. In : Proceedings of the ACM first Ph.D. workshop in CIKM, PIKM '07, pp. 41–48. ACM, New York, NY, USA (2007) [71](#)
- [132] Ma, K.L. : Parallel volume ray-casting for unstructured-grid data on distributed-memory architectures. In : Proceedings of the IEEE symposium on Parallel rendering, pp. 23–30. ACM (1995) [33](#)
- [133] MacKenzie, I.S., Kauppinen, T., Silfverberg, M. : Accuracy measures for evaluating computer pointing devices. In : Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '01, pp. 9–16. ACM, New York, NY, USA (2001). DOI <http://doi.acm.org/10.1145/365024.365028>. URL <http://doi.acm.org/10.1145/365024.365028> [92](#)
- [134] McCrickard, D.S., Kehoe, C.M. : Visualizing search results using sqwid. pp. 51–60. ACM Press (1997) [72](#)

- [135] McGrath, C., Blythe, J. : Do you see what i want you to see ? the effects of motion and spatial layout on viewers' perceptions of graph structure. *Journal of Social Structure* **5**(2) (2004) 71
- [136] Merritt, J.O. : Evaluation of stereoscopic display benefits. in : Introduction to stereoscopic displays and applications. In : SPIE The International Society for Optical Engineering (1991) 33
- [137] Misue, K., Eades, P., Lai, W., Sugiyama, K. : Layout adjustment and the mental map. In : *J. Visual Languages and Computing*, vol. 6, pp. 183–210 (1995) 71
- [138] Mithal, A.K., Douglas, S.A. : Differences in movement microstructure of the mouse and the finger-controlled isometric joystick. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '96*, pp. 300–307. ACM, New York, NY, USA (1996). DOI 10.1145/238386.238533. URL <http://doi.acm.org.gate6.inist.fr/10.1145/238386.238533> 96
- [139] Moody, J., McFarland, D., Bender-deMoll, S. : Dynamic network visualization. *American Journal of Sociology* **110**, 1206–1241 (2005) 68, 71
- [140] Moreno, J. : *Who shall survive ? washington, dc : Nervous and mental disease publishing company* (1934) 14
- [141] Munzner, T. : Exploring large graphs in 3d hyperbolic space. *IEEE Comput. Graph. Appl.* **18**, 18–23 (1998). DOI 10.1109/38.689657. URL <http://portal.acm.org/citation.cfm?id=616053.618538> 38, 52
- [142] Munzner, T. : A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics* **15**(6), 921–928 (2009) 44
- [143] Mutzel, P., Gutwenger, C., Brockenauer, R., Fialko, S., Klau, G., Krüger, M., Ziegler, T., Näher, S., Alberts, D., Ambras, D., et al. : A library of algorithms for graph drawing. In : *Graph Drawing*, pp. 456–457. Springer (1998) 15
- [144] Nancel, M., Wagner, J., Pietriga, E., Chapuis, O., Mackay, W. : Mid-air pan-and-zoom on wall-sized displays. In : *Proceedings of the 2011 annual conference on Human factors in computing systems, CHI '11*, pp. 177–186. ACM, New York, NY, USA (2011). DOI 10.1145/1978942.1978969. URL <http://doi.acm.org/10.1145/1978942.1978969> 110
- [145] Nelson, T.R., Ji, E.K., Lee, J.H., Bailey, M.J., Pretorius, D.H. : Stereoscopic evaluation of fetal bony structures. *Journal of Ultrasound in Medicine* **27**(1), 15–24 (2008) 35
- [146] Ni, T., Bowman, D.A., Chen, J. : Increased display size and resolution improve task performance in information-rich virtual environments. In : *Procee-*

- dings of Graphics Interface 2006, GI '06, pp. 139–146. Canadian Information Processing Society, Toronto, Ont., Canada, Canada (2006). URL <http://dl.acm.org/citation.cfm?id=1143079.1143102> 111
- [147] Noack, A. : An energy model for visual graph clustering. In : Proceedings of the 11th International Symposium on Graph Drawing (GD 2003), pp. 425–436. Springer (2004) 16, 78, 120
- [148] Noack, A. : Modularity clustering is force-directed layout. Phys. Rev. E **79**, 026,102 (2009). DOI 10.1103/PhysRevE.79.026102. URL <http://link.aps.org/doi/10.1103/PhysRevE.79.026102> 50
- [149] Norman, J.F., Todd, J.T., Phillips, F. : The perception of surface orientation from multiple sources of optical information. Perception & Psychophysics **57**(5), 629–636 (1995) 30
- [150] North, S. : Incremental layout in dynadag. In : Proceedings of the 3rd International Symposium on Graph Drawing, 1027, pp. 409–418. Springer (1995) 71
- [151] Osawa, N., Asai, K., Sugimoto, Y.Y. : Immersive graph navigation using direct manipulation and gestures. In : Proceedings of the ACM symposium on Virtual reality software and technology, VRST'00, pp. 147–152. ACM, New York, NY, USA (2000) 15, 39, 109
- [152] Papakostas, A., Tollis, I.G. : Incremental orthogonal graph drawing in three dimensions. In : Proceedings of the 5th International Symposium on Graph Drawing, GD '97, pp. 52–63. Springer-Verlag, London, UK (1997). URL <http://portal.acm.org/citation.cfm?id=647549.728928> 38
- [153] Parker, G., Franck, G., Ware, C. : Visualization of large nested graphs in 3d. Special issue of the Journal of Visual Languages and Computing **9**, 299–317 (1998) 38
- [154] Patrignani, M., Vargiu, F. : 3dcube : A tool for three dimensional graph drawing. In : G. DiBattista (ed.) Graph Drawing, *Lecture Notes in Computer Science*, vol. 1353, pp. 284–290. Springer Berlin / Heidelberg (1997) 51
- [155] Patterson, M.J., Rinalducci, E.J. : The effects of aerial perspective on altitude estimation. In : Proceedings of the Human Factors and Ergonomics Society Annual Meeting October, pp. 542–544 (1984) 32
- [156] Paulisch, F.N., Tichy, W.F. : Edge : an extendable graph editor. Softw. Pract. Exper. **20**(S1), 63–88 (1990). DOI 10.1002/spe.4380201307. URL <http://dx.doi.org.gate6.inist.fr/10.1002/spe.4380201307> 71

- [157] Pausch, R., Proffitt, D., Williams, G. : Quantifying immersion in virtual reality. In : Proceedings of the 24th annual conference on Computer graphics and interactive techniques, SIGGRAPH '97, pp. 13–18. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (1997). DOI 10.1145/258734.258744. URL <http://dx.doi.org/10.1145/258734.258744> 22
- [158] Perer, A., Shneiderman, B. : Integrating statistics and visualization : case studies of gaining clarity during exploratory data analysis. In : CHI '08 : Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, pp. 265–274. ACM (2008) 54
- [159] Peterson, S., Axholt, M., Ellis, S. : Label segregation by remapping stereoscopic depth in far-field augmented reality. In : Mixed and Augmented Reality, 2008. ISMAR 2008. 7th IEEE/ACM International Symposium on, pp. 143–152 (2008). DOI 10.1109/ISMAR.2008.4637341 35
- [160] Peterson, S., Axholt, M., Ellis, S. : Managing visual clutter : A generalized technique for label segregation using stereoscopic disparity. In : Virtual Reality Conference, 2008. VR '08. IEEE, pp. 169–176 (2008). DOI 10.1109/VR.2008.4480769 35
- [161] Pinaud, B., Kuntz, P. : GVSR : an On-Line Guide for Choosing a Graph Visualization Software. In : U. Brandes, S. Cornelsen (eds.) 18th International Symposium on Graph Drawing, LNCS, vol. 6502, pp. 400–401. Springer, Constance, Allemagne (2011). URL <http://hal.inria.fr/inria-00516580> 56
- [162] Pylyshyn, Z. : The role of location indexes in spatial perception : A sketch of the first spatial-index model. *Cognition* **32**(1), 65–97 (1989) 23
- [163] Reda, K., Febretti, A., Knoll, A., Aurisano, J., Leigh, J., Johnson, A., Papka, M., Hereld, M. : Visualizing large, heterogeneous data in hybrid-reality environments. *Computer Graphics and Applications, IEEE* **33**(4), 38–48 (2013). DOI 10.1109/MCG.2013.37 109, 111, 117
- [164] Reddy, P.K., Kitsuregawa, M., Sreekanth, P., Rao, S.S. : A graph based approach to extract a neighborhood customer community for collaborative filtering. In : Databases in Networked Information Systems, pp. 188–200. Springer (2002) 76
- [165] Ressler, S. : Social network analysis as an approach to combat terrorism : Past, present, and future research. *Homeland Security Affairs* **2**(2), 1–10 (2006) 14
- [166] Robertson, G.G., Mackinlay, J.D., Card, S.K. : Cone trees : animated 3d visualizations of hierarchical information. In : Proceedings of the SIGCHI

- conference on Human factors in computing systems, pp. 189–194. ACM (1991) [51](#)
- [167] Rosenbaum, A.E., Huda, W., Lieberman, K.A., Caruso, R.D. : Binocular three-dimensional perception through stereoscopic generation from rotating images. *Academic Radiology* **7**(1), 21–26 (2000) [33](#), [37](#)
- [168] Rosenberg, A.L. : Three-dimensional vlsi : a case study. *J. ACM* **30**, 397–416 (1983). DOI <http://doi.acm.org/10.1145/2402.322384>. URL <http://doi.acm.org/10.1145/2402.322384> [49](#), [51](#)
- [169] Rosenstiehl, P. : Claude berge, ses graphes et hypergraphes. *Mathématiques et sciences humaines. Mathematics and social sciences* (160) (2002) [14](#)
- [170] Rosset, A., Spadola, L. : Osirix : an open-source software for navigating in multidimensional dicom images. *Journal of Digital Imaging* **17**(3), 205–216 (2004) [33](#)
- [171] Rosvall, M., Bergstrom, C.T. : Mapping change in large networks. *PLoS ONE* **5**(1) (2010) [70](#)
- [172] Roy, W.G. : The unfolding of the interlocking directorate structure of the united states. *American Sociological Review* (1983) [71](#)
- [173] Ruiz, J., Tausky, D., Bunt, A., Lank, E., Mann, R. : Analyzing the kinematics of bivariate pointing. In : *Proceedings of graphics interface 2008, GI '08*, pp. 251–258. Canadian Information Processing Society, Toronto, Ont., Canada, Canada (2008). URL <http://dl.acm.org/citation.cfm?id=1375714.1375756> [92](#)
- [174] Sarkar, M., Brown, M.H. : Graphical fisheye views of graphs. In : *CHI '92 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 83–91. ACM (1992) [15](#), [53](#)
- [175] Schild, J., LaViola, J., Masuch, M. : Understanding user experience in stereoscopic 3d games. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12*, pp. 89–98. ACM, New York, NY, USA (2012). DOI [10.1145/2207676.2207690](http://doi.acm.org/10.1145/2207676.2207690). URL <http://doi.acm.org/10.1145/2207676.2207690> [36](#)
- [176] van Schooten, B.W., van Dijk, E.M.A.G., Zudilova-Seinstra, E., Suinesiaputra, A., Reiber, J.H.C. : The effect of stereoscopy and motion cues on 3d interpretation task performance. In : *Proceedings of the International Conference on Advanced Visual Interfaces, AVI '10*, pp. 167–170. ACM, New York, NY, USA (2010). DOI <http://doi.acm.org.gate6.inist.fr/10.1145/1842993.1843023>. URL <http://doi.acm.org.gate6.inist.fr/10.1145/1842993.1843023> [31](#), [36](#)

- [177] Shannon, C.E. : A mathematical theory of communication. Bell System Technical Journal **27**, 379–423 (1948) [102](#)
- [178] Shannon, P., Markiel, A., Ozier, O., Baliga, N.S., Wang, J.T., Ramage, D., Amin, N., Schwikowski, B., Ideker, T. : Cytoscape : A software environment for integrated models of biomolecular interaction networks. Genome Research **13** (2003) [38](#)
- [179] Shivani, H.A., Pollett, C.B., Stamp, M.C. : On using mouse movements as a biometric. In : Proceedings of the International Conference on Computer Science and its Applications (2005) [92](#), [93](#)
- [180] Shneiderman, B. : Tree visualization with tree-maps : 2-d space-filling approach. ACM Transactions on graphics (TOG) **11**(1), 92–99 (1992) [47](#)
- [181] Shneiderman, B. : The eyes have it : a task by data type taxonomy in information visualization. In : IEEE Symp. on Visual Languages. IEEE Press (1996) [47](#)
- [182] Shneiderman, B. : Treemaps for space-constrained visualization of hierarchies (1998) [47](#)
- [183] Shoemake, K. : Arcball : a user interface for specifying three-dimensional orientation using a mouse. In : Proceedings of the conference on Graphics interface '92, pp. 151–156. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1992). URL <http://dl.acm.org/citation.cfm?id=155294.155312> [65](#), [77](#)
- [184] Smith, M.A., Shneiderman, B., Milic-Frayling, N., Mendes Rodrigues, E., Barash, V., Dunne, C., Capone, T., Perer, A., Gleave, E. : Analyzing (social media) networks with nodexl. In : Proceedings of the fourth international conference on Communities and technologies, pp. 255–264. ACM (2009) [44](#)
- [185] Sollenberger, R., Milgram, P. : The effects of stereoscopic and rotational displays in a three-dimensional path-tracing task. In : Human Factors, vol. 35, pp. 483–500 (1993) [38](#)
- [186] Sutherland, I. : The ultimate display. In : Proceedings of IFIPS Congress 1965, New York, May 1965, Vol. 2, pp. 506-508 (1965) [22](#)
- [187] Sylvester, J.J. : Chemistry and algebra. Nature **17**, 284 (1878) [14](#)
- [188] Tang, L., Liu, H. : Community Detection and Mining in Social Media. Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool Publishers (2010) [46](#)
- [189] Teichner, W.H., Kobrick, J.L. : The effects of terrain and observation distance on relative depth perception. American Journal of Psychology **68**, 193–208 (1955) [28](#)

- [190] Telea, A., Ersoy, O. : Image-based edge bundles : Simplified visualization of large graphs. In : Proceedings of Eurovis'10 (2010) [15](#), [53](#)
- [191] Thomas, J.J., Cook, K.A. : Illuminating the path : The research and development agenda for visual analytics. IEEE Computer Society Press (2005) [42](#)
- [192] Trier, M. : Towards dynamic visualization for understanding evolution of digital communication networks. Information Systems Research **19**(3) (2005) [68](#)
- [193] Tutte, W. : How to draw a graph. In : London Mathematical Society, vol. 13, pp. 743–768 (1964) [15](#), [43](#)
- [194] Van Wijk, J.J. : The value of visualization. In : Visualization, 2005. VIS 05. IEEE, pp. 79–86. IEEE (2005) [42](#)
- [195] Veldhuizen, T.L. : Dynamic multilevel graph visualization. CoRR **abs/0712.1549** (2007) [38](#)
- [196] Wade, N. : Charles wheatstone (1802-1875). Perception **31**(2), 265 – 272 (2002) [16](#)
- [197] Wang, X., Miyamoto, I. : Generating customized layouts. In : Graph Drawing, pp. 504–515. Springer (1996) [50](#)
- [198] Wang, X.H., Durick, J.E., Herbert, D.L., Golla, S.K.L.A., Shinde, D.D., Piracha, S., Foley, K., Fuhrman, C.R., Shindel, B.E., Leader, J.K., Good, W.F. : Compare display schemes for lung nodule ct screening. Journal of Digital imaging **24**(3), 478–484 (2010) [36](#)
- [199] Wardle, S.G., Cass, J., Brooks, K.R., Alais, D. : Breaking camouflage : Binocular disparity reduces contrast masking in natural images. Journal of Vision **10**(14) (2010). DOI 10.1167/10.14.38. URL <http://www.journalofvision.org/content/10/14/38.abstract> [33](#)
- [200] Ware, C., Franck, G. : Evaluating stereo and motion cues for visualizing information nets in three dimensions. In : ACM Transactions on Graphics, vol. 15, pp. 121–139 (1996) [38](#)
- [201] Ware, C., Mitchell, P. : Reevaluating stereo and motion cues for visualizing graphs in three dimensions. In : Proceedings of the 2nd Symposium on Applied Perception in Graphics and Visualization, APGV 2005, vol. 95. ACM (2005) [31](#)
- [202] Ware, C., Mitchell, P. : Visualizing graphs in three dimensions. In : ACM Transactions on Applied Perception, vol. 5, pp. 2–15 (2008) [38](#)
- [203] Ware, C., Purchase, H., Colpoys, L., McGill, M. : Cognitive measurements of graph aesthetics. Information Visualization **1**(2), 103–110 (2002) [48](#)

- [204] Weiser, M. : The computer for the 21st century. *ACM SIGMOBILE Mobile Computing and Communications Review* **3**(3), 3–11 (1999). DOI 10.1145/329124.329126. URL <http://doi.acm.org/10.1145/329124.329126> 108
- [205] Wilcoxon, F. : Individual Comparisons by Ranking Methods. *Biometrics Bulletin* **1**(6), 80–83 (1945). DOI 10.2307/3001968. URL <http://dx.doi.org/10.2307/3001968> 98
- [206] Witmer, B.G., Singer, M.J. : Measuring presence in virtual environments : A presence questionnaire. *Presence : Teleoperators and Virtual Environments* **7**(3), 225–240 (1998). DOI 10.1162/105474698565686. URL <http://dx.doi.org/10.1162/105474698565686> 22
- [207] Wood, D.R. : An algorithm for three-dimensional orthogonal graph drawing. pp. 332–346 (1998) 51
- [208] Xia, J., Ip, H.H., Samman, N., Wang, D., Kot, C.S., Yeung, R.W., Tideman, H. : Computer-assisted three-dimensional surgical planning and simulation : 3d virtual osteotomy. *International journal of oral and maxillofacial surgery* **29**(1), 11–17 (2000) 110
- [209] Yang, Y., Wurtele, E.S., Cruz-Neira, C., Dickerson, J.A. : Hierarchical visualization of metabolic networks using virtual reality. In : *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications, VRCIA '06*, pp. 377–381. ACM, New York, NY, USA (2006) 39
- [210] Zigelbaum, J., Browning, A., Leithinger, D., Bau, O., Ishii, H. : g-stalt : a chirocentric, spatiotemporal, and telekinetic gestural interface. In : *TEI '10 : Proceedings of the 4th International Conference on Tangible and Embedded Interaction*, pp. 261–264. ACM (2010). URL <http://doi.acm.org/10.1145/1709886.1709939> 110

Thèse de Doctorat

Nicolas GREFFARD

Visualisation Stéréoscopique et Interactive de Structures en Communautés dans des Graphes

Stereoscopic and Interactive Visualization of Community Structures in Graphs

Résumé

Depuis les premiers travaux de Moreno en 1934, l'analyse de réseaux sociaux s'est toujours accompagnée de diagrammes représentant les relations entre individus. Depuis, la visualisation de graphes n'a cessé de se développer au sein des communautés de visualisation d'information et de dessin de graphes. Au-delà des travaux se concentrant sur les problèmes combinatoires et algorithmiques posés par les dessins de graphes on parle désormais de fouille visuelle de réseaux et plus généralement de visual analytics en intégrant l'utilisateur au cœur de l'analyse. Dans cette thèse nous nous concentrons sur les supports visuels interactifs stéréoscopiques qui permettent de piloter le processus de fouille. En utilisant un environnement expérimental *ad hoc*, nous essayons d'en évaluer l'impact dans une tâche très populaire de détection de structures en communautés. A travers différentes expériences, nous montrons que pour une classe de graphes particulière la 2D semble plus adapté pour les graphes simples tandis que la 3D stéréo est bénéfique pour les graphes les plus complexes. Nous identifions également des différences dans l'usage des interactions entre la mono et la stéréo, ce qui semble illustrer des différences de comportement caractérisant des différences de stratégie d'utilisation entre ces deux conditions. Nos travaux se prolongent sur le plan technologique par le développement d'une bibliothèque permettant des interactions "mains-libres" adaptées à la fouille visuelle debout devant un grand écran.

Mots clés

Visualisation, Graphes, Réseaux Sociaux, Communautés, 3D Stéréoscopique, Fouille Visuelle

Abstract

Ever since the pioneering work of Moreno in 1934, social network analysis has always included drawings depicting relationships between actors. From these days, the graph visualization field has grown within the graph drawing and information visualization communities. Besides the algorithmic and combinatoric questions arising from graph drawings, new challenges now include networks visual data mining. Usually referred to as visual analytics it involves the integration of the user at the heart of the analysis. In this thesis, we focus on interactive and stereoscopic visual restitutions allowing the user to drive the mining process. Using a *ad hoc* experimental environment, we try to assert their its impact on a popular task of community detection. Through several experiments, we show that for a specific class of graphs, 2D seems more adapted for the easier graphs while stereoscopic 3D is beneficial for the more complex ones. We also identify some differences in the interactions between the stereo and mono conditions, which seems to indicate behavioral differences emerging from differing interaction strategies. We also propose some prospects such as the implementation of a library allowing hands-free interactions adapted to visual mining in front of a large screen.

Key Words

Visualization, Graphs, Social Networks, Communities, Stereoscopic 3D, Visual Analytics

